MULTIPLE MODEL ADAPTIVE ESTIMATION AND CONTROL REDISTRIBUTION PERFORMANCE ON THE VISTA F-16 DURING PARTIAL ACTUATOR IMPAIRMENTS

Volume I

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/ENG/97D-23

MULTIPLE MODEL ADAPTIVE ESTIMATION AND CONTROL REDISTRIBUTION PERFORMANCE ON THE VISTA F-16 DURING PARTIAL ACTUATOR IMPAIRMENTS

Volume 1

THESIS

Curtis Steven Clark

AFIT/GE/ENG/97D-23

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government

AFIT/GE/ENG/97D-23

MULTIPLE MODEL ADAPTIVE ESTIMATION AND CONTROL REDISTRIBUTION PERFORMANCE ON THE VISTA F-16 DURING PARTIAL ACTUATOR IMPAIRMENTS

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Electrical Engineering

Curtis Steven Clark, B.S. Engineering-Physics

December, 1997

Acknowledgements

I would like to take this opportunity to thank those persons and institutions which mean so much to me. I would like to thank my parents, Nelson and Carol Clark, for teaching me, by example, the value of hard work and the importance of truth and honesty. I would like to thank my wonderful wife, Janice, for her patience, her understanding, and for recently giving me a beautiful son, Coleman. I truly have everything a person could want, and furthermore, I am lucky enough to realize it. I would like to express my gratitude to Dr. Peter Maybeck, who has inspired me not only to be my professional best, but to put forth my best personal qualities as well. A sincere, special thanks goes to Lt. Mike Stepaniak, for being so generous with his time, and ensuring that I got thoroughly acquainted and comfortable with the simulation and research environment. Finally, I would like to express my appreciation of and fondness for the institution of learning called AFIT. It has been a sincere pleasure and a privilege to have come to AFIT, to have attended classes with future test pilots, future astronauts, and future Air Force leaders, and to have been instructed by some of the best, the most dedicated, and the most noteworthy professionals in their fields, Dr. Maybeck in especial. I only hope that despite the current national trend of cultural and intellectual deconstruction, AFIT will survive with its high standards intact, to teach, to inspire, and to reinvigorate engineering professionals as it has done for me.

Curtis Steven Clark

December 1997

Table of Contents

VOLUME I

VOLUME II

VOLUME III

List of Figures

List of Tables

Page

Abstract

Multiple Model Adaptive Estimation with Control Reconfiguration (MMAE/CR) capability to estimate and compensate for *partial* actuator failures, or "impairments" is investigated using the high-fidelity, nonlinear, six-degree-of-freedom, VISTA F-16 simulation which currently resides on the Simulation Rapid-Prototyping Facility (SRF). After developing a model for inserting partial actuator impairments into the VISTA F-16 truth model, research begins with a battery of single actuator impairment tests. This stage of research explores the capability of the existing MMAE algorithm to estimate *single*, partial actuator impairments, and helps to define refinements and expansions needed in the MMAE algorithm for the second phase of research: the detection and estimation of *dual*, total and partial actuator impairments. It is seen from the first stage of research that, while MMAE *is* able to estimate partial impairments, there are refinements needed, such as "probability smoothing and quantization", to compensate for the quality of MMAE probability data and to provide a better, more stable estimate value to the Control Reconfiguration module. The Kalman filters and the dual, partial failure filter banks necessary for the detection of dual, partial actuator impairments are also defined as a result of the single impairment tests. Fifteen more banks of "partial first-failure" Kalman filters are added to the existing MMAE algorithm, as well as the "bank swapping" logic necessary to transition to them. Once the revised and expanded MMAE/CR algorithm is ready, research begins on dual combinations of total and partial actuator impairments. While results of these tests (for other than total impairments) are not as good as originally hoped or expected, the potential for better performance is evident. Of equal importance, a fuller understanding is gained of the reasons for lackluster MMAE performance, and of the remedial actions necessary to improve results in future research.

MULTIPLE MODEL ADAPTIVE ESTIMATION AND CONTROL REDISTRIBUTION PERFORMANCE ON THE VISTA F-16 DURING PARTIAL ACTUATOR IMPAIRMENTS

1. Introduction

1.1. Chapter Overview

This thesis presents an investigation of a Multiple Model Adaptive Estimation and Control Reconfiguration (MMAE/CR) Algorithm's ability to estimate and compensate for partial, or "soft" actuator impairments. This chapter explains the motivation behind MMAE/CR in Section 1.2, and Section 1.3 will present a problem statement and define the research goals. Basic research assumptions are enumerated in Section 1.4, followed in Section 1.5 by an outline of the thesis format. The chapter concludes with a summary.

1.2. Motivation

The United States Air Force has recognized the need for fault-tolerant flight control systems for many years. Flight control systems which are able to adapt and compensate for potentially hazardous changes in air vehicle configuration, due, for instance, to battle damage, sensor failures, or actuator failures, are feasible in the age of the modern "fly-by-wire" flight control system. In fact, compensating, or "reconfiguring" for degraded or failed flight control system components is relatively easy *once the identification and severity of the failure is known.* It is to this identification problem that this thesis research effort will continue the application of Multiple Model Adaptive Estimation, or MMAE.

MMAE has demonstrated its suitability for the task in prior research efforts [14, 26, 45], by rapidly and correctly identifying *total* failures of sensors and/or actuators on board a high fidelity simulation of the VISTA F-16. This exceptional detection performance by MMAE enabled the cascaded use of Control Reconfiguration (CR) to restore all, or most of, the aircraft control authority lost during the failure, and to do it rapidly. These results have naturally led to the question that this thesis will address: "Is it feasible to apply MMAE to the task of detecting and compensating for *partial* actuator failures?"

1.3. Problem Statement

This thesis research will explore the ability of the current Multiple Model Adaptive Estimation (MMAE) scheme to detect and estimate *partial* actuator failures, or "impairments". To accomplish this, a model of a "partial actuator impairment" and the means of inserting it into the VISTA F-16 simulation will be defined. This model should, for total actuator impairments, duplicate the results of earlier research efforts [26, 45]. A series of single, partial actuator impairment tests will then be performed to evaluate how well the MMAE can "blend" the outputs of its Kalman Filters to identify and to estimate the degree of impairment accurately. The results of these tests are extremely important, and will be used to decide 1) whether MMAE needs modifications to improve performance, and, 2) on an appropriate interface scheme with the Control Reconfiguration module, to include all expansions of the MMAE algorithm necessary for the study of *dual* impairments. Research will then be directed towards the study of MMAE detection and estimation performance in cases of *dual, partial* actuator impairments and "partial actuator, *total* sensor" impairments (since *partial sensor* impairments are not modeled at this time). These tests will provide the data necessary to decide on further modifications to MMAE, and on the most promising courses for future research.

1.4. Assumptions

It is assumed that the truth model of the VISTA F-16 is unchanged from the time of its delivery from General Dynamics, and hence that it is still an accurate model of actual F-16 flight characteristics in the flight regime studied in this thesis. It is also assumed that the existing linear, time-invariant models used within the MMAE algorithm are accurate and adequate representations of the F-16 truth model, at the selected Mach number and altitude, for the purposes for which they are employed.

1.5. Thesis Format

Chapter 1 has provided the reader with the motivations directing this thesis research, and has stated the major assumptions made before beginning the research. Chapter 2 begins with a summary of previous research pertinent to this application of MMAE, including theoretical developments as well as results obtained in prior theses. A mathematical explanation of the MMAE algorithm, as it relates to this thesis application, is offered next, followed by a description of algorithm refinements and extensions made by prior research efforts. Chapter 3 describes the VISTA F-16 truth model in greater detail, and follows with specific, in-depth descriptions and "build-up" of the mathematical models used within the MMAE algorithm. Control Redistribution, or "Reconfiguration" is described next, including historical developments along with mathematical development of the concept. Chapter 4 details the simulation results of single and dual, partial impairments. Single impairment results are presented first, followed by complete details and explanations of modifications to and expansions of the MMAE algorithm necessary for the study of dual, partial impairments, the results of which are presented last. Chapter 5 presents conclusions reached in this thesis, and recommendations for future thesis research. Appendices provide additional, relevant data and information necessary for follow-on research efforts.

1.6. Chapter Summary

This chapter provided the reader with a motivation for undertaking this thesis research and stated the research problem this thesis addresses. Major assumptions made at the outset of research were stated, followed by an introduction to the format this thesis will follow.

2. *MMAE: History and Algorithm Development*

2.1. *Chapter Overview*

This chapter begins with a chronological summary of previous work in Multiple Model Adaptive Estimation (MMAE) and Multiple Model Adaptive Control (MMAC) leading up to this research. Section 2.4 explains pertinent theory and describes the MMAE algorithm, starting with the Kalman filters within the MMAE and proceeding to the calculation of Conditional Hypothesis Probabilities, which provide the actual adaptation mechanism for MMAE. Modifications found necessary during past research for better algorithm performance are explained, as is the extension of basic MMAE to a "hierarchical structure" to handle multiple impairments. The chapter is summarized in Section 2.9.

2.2. *Chronological Development of Multiple Model Adaptive Estimation and Multiple Model Adaptive Control (MMAE/MMAC)*

The idea of Multiple Model Adaptive Estimation was presented for the first time in 1965 by Magill [27], in which he presented the concept of blending weighted estimates from individual filters, each based on a hypothesized parameter realization, to form what was called the "optimal adaptive estimate". While not specifically named MMAE, the structure of using a bank of parallel Kalman estimators was put into place.

Athans and Chang [9] contributed significantly to the practical implementation of MMAE algorithms using distributed computation, and Lainiotis [24] developed the idea of Multiple Model Adaptive Control (MMAC) by cascading each estimator in the MMAE structure with an optimal controller matched to that estimator, yielding a parallel bank of "optimal" *controllers* to generate system control inputs, which would then be blended (like MMAE estimates) into an "optimal adaptive control" signal. Like the MMAE, if the "true" optimal control signal was actually between two filters, or "channels" in an operating MMAC algorithm, then the MMAC algorithm would blend control signals weighted most heavily

towards adjacent channels on either side of that true value. Lainiotis also found that, due to a recursive method of calculation, a probability used to weight a filter estimate would shut down for all time once a zero value was reached, henceforth "locking out" that specific channel.

A.S. Willsky [50] concluded in a 1976 comparison of various Failure Detection and Isolation (FDI) schemes that multiple model methods were the best all-around performers in terms of speed and accuracy for detecting the widest class of failures, and this announcement no doubt greatly influenced FDI research in the direction of multiple model methods. Greene and Willsky [19, 20] further investigated the performance characteristics of MMAC algorithms.

An important work, introducing ideas used since in MMAE research, came in 1977 by Athans et. al. [1]. That research effort was an attempt to introduce and flight test "next generation" ideas in flight control, specifically MMAC, on board an F-8 "Crusader" fighter jet. Looking to replace conventional methods of scheduling flight control system gains, an MMAC was used to estimate flight condition given several discrete possibilities, then blending outputs of the corresponding controllers to calculate the resulting longitudinal control signal. Though experimental conditions and on-board computer power were overly restrictive, and results were not directly relevant to the next generation of unstable and highly augmented fighter aircraft, several ideas used in this research were introduced. The idea of putting a minimum bound on allowed probability (Section 2.8.1) to prevent probability calculations from "latching" at zero for all time kept all filters in the MMAC bank responsive to time-varying parameter changes. The concept and ill effects of "Beta Dominance" (Section 2.8.2) was introduced, as was the proposed need for small excitation or "dither" signals to aid MMAC convergence by creating residual activity in an otherwise quiescent environment or flight condition.

Chang, Athans, and Cox [9, 10, 13] proved that the blended estimates and controls as output by MMAE and MMAC algorithms were not optimal as previously thought, except in cases of constant unknown parameters for which the "true" but unknown parameter value matched one of the filter hypotheses perfectly. If the true parameter does not match a hypothesis, then the blended estimate and control signal from MMAE/MMAC converges to the discrete point nearest to the true parameter value in the Baram distance measurement sense [2,3,4]. These suboptimal qualities are to be accepted, since the computational burden to achieve true optimality is prohibitive [9], and the MMAE scheme as implemented

does provide quick, accurate, and "close-to-optimum" estimates, even between widely spaced filter hypotheses.

Fry and Sage [16] introduced the concept of the hierarchical filter structure, used in this research to reduce computational loading in the face of multiple failures. To cover the entire range of possible single and double sensor and actuator *total* failures, or "*impairments*", in this thesis would require a single parallel bank of at least 67 parallel filter-estimators. Using the hierarchical concept, the 67 parallel, on-line filters are reduced to a bank of 12 parallel filters on line at any one time (Section 2.8.5).

MMAE/MMAC Research at AFIT

The U.S Air Force Institute of Technology (AFIT) has made numerous contributions to the state of the art both in MMAE/MMAC research and in application of MMAE/MMAC technology to solve problems specific to the Air Force. In the field of flight control for instance, the Air Force has long since identified a need for *reliable* fault detection and isolation (FDI) algorithms as a "front end" to fault-tolerant, or *robust*, flight control systems which increase reliability, maintainability, and survivability of combat aircraft.

The "β dominance" discussed by Athans [1] had the undesired effect of causing false declarations of sensor failure in the MMAE algorithm (Section 2.8.2). Maybeck and Suizu [30], in a 1985 paper, suggested that the β term could be stripped from all conditional probability calculations [14, 26, 31, 33, 40, 45]. Because the same β term appeared in both the numerator and denominator of the probability calculations (to be discussed in detail in Section 2.8.2), and since the β term contained no useful information about failures anyway, the conditional probability calculations would still be *approximately* valid, and false alarms on sensors would be reduced or even eliminated if the β terms were removed. This modification, used in this thesis, was made and the predicted effects verified in later research [33, 46].

In some applications of MMAE/MMAC, even the hierarchical filter structure can get too large to be practical, due to the sheer size of the parameter, or hypothesis, space which must be covered. Maybeck and Hentz [29] suggested "sweeping", or searching the parameter space, much like a radar searching for and locking onto a target, by using a bank of filters based upon variable ("moving") hypotheses. Once finding

an area of parameter space yielding significant non-zero probability activity, the bank of variable-hypothesis filters would become a standard bank of filters based on stationary hypotheses like those used in this thesis.

2.3. *Applications of MMAE/MMAC Specific to Flight Control.*

MMAE/MMAC is proving applicable to a wide field of problems, but one of the most fruitful areas of research has been applying MMAE/MMAC to explore flight control problems of critical interest to the Air Force. Under the auspices of Wright Laboratory, AFIT has taken a lead in this research area for several years.

Pogoda in 1988 [44] investigated the ability of a Multiple Model Adaptive Control scheme to compensate for total loss of a single actuator, sensor, or "pseudo surface" actuator on a linear, longitudinal model of the STOL F-15 on landing approach. In spite of the effects of β dominance, which caused a false detection of sensor failure, results were promising in that control was maintained, and valuable experience with MMAC was gained, especially insight into the importance of correct filter tuning to optimize MMAC performance.

Where Pogoda's research concentrated on *total* losses of a *single* actuator or sensor, Stevens [33, 46] in 1989 explored MMAC performance during *partial as well as total* failures of *single and dual* actuators and/or sensors. Stevens made a major correction to the MMAC algorithm, and introduced several modifications critical to future research. First, it was decided to change the control input to each elemental Kalman filter, $\mathbf{u}_k$, to the blended control signal, $\mathbf{u}_{MMAC}$. By giving the filters the *true* control signal, residual behavior and MMAC algorithm performance were enhanced so well that Stevens was able to handle all single impairment cases. To handle the dual failure cases, Stevens implemented the hierarchical filter structure. The β term, troublesome to Pogoda's research, was eliminated (and in all future research as well), vastly improving MMAC detection performance in both speed and ability to stay "locked on" to the correct failure without false declarations of sensor failures. Stevens' results for multiple failures verified the hierarchical filter structure's ability to switch between filter banks rapidly enough so that stability was

maintained. His results for partial failures demonstrated the ability of MMAC to blend the outputs of adjacent filters, in cases of failures not modeled exactly by any one filter, to obtain estimates agreeing with the "true" impairment. Stevens' results are of extreme importance to this thesis researching MMAE with Control Reconfiguration. Since this thesis will use the hierarchical filter structure and will use blended conditional probability values to carry out calculations within the Control Reconfiguration scheme, probability convergence speed and filter switching speed are of the utmost importance.

Martin's research in 1990 [28] implemented MMAC for adaptive control in both longitudinal and lateral directional channels. Where research to date had been using a *stable*-airframe, longitudinal model of an aircraft at one flight condition, Martin implemented MMAC on an aircraft model more representative of current, highly augmented, *unstable* fighter aircraft, and at several points throughout the flight envelope.

Menke [39, 40] moved research in a slightly different direction, away from MMAC and toward MMAE-based control. Because extensive aerodynamic and flight control system simulation data was available for validation purposes, the truth model was changed to the VISTA F-16, linearized at a flight condition and c.g. location yielding an unstable linear airframe model requiring stability augmentation. This enabled testing and evaluating MMAE-based control in a more challenging environment "in the loop" with a full-scale flight control system: certainly a prerequisite for moving MMAE/MMAC research away from academic exercises and toward a future flight test. Test points covered single and dual, *complete* failures of actuators and/or sensors as well as single and dual, *partial* impairments of actuators and sensors. Research examined primarily the ability of MMAE to detect and isolate failures quickly and accurately, as well as the speed and accuracy of state estimation. The use of "dithering" the control surfaces to excite system states and to speed up failure detection during benign periods of flight was investigated, as was the use of scalar residual monitoring (examining individual components of each filter's residual vector; this had also been considered by Stevens [46] previously) to provide second corroborative votes on some failures. MMAE performed well, detecting and isolating all single and most double failures. Menke's scope of research on the linear VISTA F-16 simulation (with nonlinear actuator rate and position limits) is very close to the scope of this thesis, and hence his contributions and results are extremely important.

Up to the time of Eide's [14] research in 1994, all MMAE/MMAC research had been performed using linear, time invariant (LTI) models of the host aircraft as truth models. Where Menke studied MMAE

detection performance on a linear model of the VISTA F-16 (with actuator position and rate limits) and got very promising results, Eide took Menke's research a step further and integrated MMAE with the non-linear, six degree-of-freedom, General Dynamics VISTA F-16 simulation as implemented on the Simulation Rapid Prototyping Facility, or SRF [23]. This simulation, received from General Dynamics (now Lockheed) in 1989, and verified in the normal, subsonic flight regime by check cases and in real-time flight test by F-16 pilots [12], allowed Eide to duplicate and expand prior research on a high-fidelity truth model. Eide explored detection of all possible single and dual complete (but not partial) failures of actuators, sensors, and actuator-sensor combinations, and after tuning the MMAE filters properly, got nearly perfect detection of all single actuator and sensor failures [14]. Results for dual complete failures were reported as very good overall, and causes of "trouble spots" which did occur were studied and understood. Eide proved that, with proper tuning, MMAE detected dual failures almost as well as single failures. He also implemented and used control surface dithering as proposed by earlier work, and explored scalar residual monitoring to enhance detection of sensor failures. Due to the increased capability of the new VISTA truth model, Eide also studied MMAE performance in off-design flight conditions. Finding that MMAE performance deteriorates rapidly for very small dynamic pressure variations due to model mismatch within MMAE filters, Eide suggested that real-world implementations of MMAE would require gain scheduling (as a function of altitude and Mach number, or of dynamic pressure) within the elemental filters. Eide's research and results set the course of future research. MMAE *detection* capability for single and dual, total failures having been proven, what remained was the study of MMAC or MMAE-based *control* after single and dual impairments, as well as the study of detection and control after single and dual partial impairments.

After Eide's excellent results for detection of single, total impairments, Stepaniak [45] elected to further MMAC/MMAE research by studying Multiple Model Adaptive Estimation and Control (MMAE/MMAC) under conditions of total, single actuator or sensor failures. Unfortunately, Stepaniak was forced to abandon the MMAC approach due to design, numerical and modeling problems and instead rediscovered, developed, and applied what he termed "Control Redistribution". Control Redistribution (CR), as independently developed and implemented by Stepaniak, is a perfect match to the excellent and demonstrated detection capabilities of MMAE. It has the very desirable quality of being a passive, "drop-in" module to the existing, flight-proven, F-16 Block 40 flight control system: it simply takes surface

commands issued by the F-16's flight control system and uses a simple mathematical concept to redirect those commands away from failed actuators and toward unfailed actuators to restore lost control power. Furthermore, the Control Redistribution concept has an extensive, albeit forgotten, research history behind it as well, but under other names (Section 3.11.1). As in Eide's research, the detection, or MMAE stage of MMAE/CR, detected single, hard (total) failures very quickly and the Control Reconfiguration stage compensated for the loss of control capability extremely well, even at the demanding "low dynamic pressure" flight condition studied. Stepaniak also provided very insightful comments and warnings as to possible limitations of the MMAE/CR scheme, some of which had indeed been encountered previously (Section 3.11.1). Stepaniak improved upon Eide's control surface dithering scheme, and recommended that future research examine MMAE/CR performance in the face of partial and total, dual impairments of actuators, sensors, or actuator/sensor combinations.

Following Stepaniak's recommendations, Lewis [26] examined MMAE/CR performance during dual, total failures of actuators, sensors, and actuator-sensor combinations. The research was carried out in three steps. First, since Lewis used the improved dithering scheme of Stepaniak, it was necessary to repeat Eide's dual impairment test cases with the new dithering scheme and without Control Redistribution, to establish any performance benefits. Results showed significant improvement in second failure detection for over 50% of test cases when compared to Eide's original cases. The second stage of research repeated the battery of test cases once again, but now with Control Redistribution in addition to the improved dithering scheme. Dual failure, "second detection" results were terrible initially, due to rate saturation of the remaining unimpaired but overburdened actuators. Experiments with scaling the Control Redistribution commands to prevent actuator saturation yielded much better results for detecting the second failure. In stage three, a study was performed comparing the VISTA F-16 response to maneuver inputs with and without Control Reconfiguration. His results showed that Control Reconfiguration did indeed restore control power after one or even two impairments, and in at least one case, prevented aircraft departure from controlled flight.

This research, then, focuses on evaluating MMAE-based control with Control Redistribution (MMAE/CR) performance when confronted with partial failures of single and dual actuators, often referred to as "soft failures". Due to the method of simulating "partial actuator failure" in this thesis, namely by

attenuating actuator positions of the truth model by an *effectiveness factor*, $\{\varepsilon\ ;\ 0 \le \varepsilon \le 1\}$ (Section 4.3), "partial actuator failure" is termed more appropriately "a partial actuator *impairment*", or "partial loss of actuator effectiveness". Test cases take place at the same flight condition (Mach=.4, Altitude=20000 feet) as previous research, and begin with a battery of single surface partial impairments of 50% (effectiveness, $\varepsilon = .5$), 75% ($\varepsilon = .25$), 90% ($\varepsilon = .1$), and 100%. The intent of the tests at these impairment conditions is to explore the ability of the existing MMAE algorithm to blend correct probability hypotheses to produce an estimate of partial actuator impairment, and to indicate the expansion and modification of the MMAE necessary for the dual impairment studies which follow. Once the required modifications are complete, dual, *partial* actuator impairments, of all possible two-actuator impairment combinations occurring one second apart, are run and performance of the expanded MMAE/CR algorithm is evaluated. All simulations are performed *without* aircraft maneuvers, but *with* control dithering activated, since it is apparent from earlier research that, while MMAE requires dither signals for detection purposes, special measures (gain scheduling) will have to be taken to enhance MMAE performance during maneuvering flight. Results, conclusions, and recommendations are then presented.

2.4. Multiple Model Adaptive Estimation (MMAE)

In this section, the Multiple Model Adaptive Estimation algorithm is discussed in detail. We build the MMAE algorithm up, starting with equations of the Kalman filter in Section 2.5. Conditional probability calculations used to weight the Kalman filter estimates are developed next, and then both are combined to obtain the MMAE algorithm. Modifications to the MMAE algorithm made in previous research for better performance are explained in Sections 2.8.1 through 2.8.3, followed by a discussion of "dithering" in Section 2.8.4. The discussion concludes in Section 2.8.5 with the extension of MMAE to a hierarchical filter structure used in cases of multiple surface impairments.

2.5. *The Kalman filter.*

To understand the concept of Multiple Model Adaptive Estimation (MMAE), it is instructive to review the operation of its basic component, the Kalman filter. A Kalman filter takes sensor measurements and control inputs and estimates system quantities, or states of interest. To do this, the Kalman filter must have "knowledge", in the form of mathematical models, of how the system behaves and how states combine to form measurements, as well as statistical knowledge of uncertainties, or "noises", present in those models. If the mathematical models upon which a Kalman filter is based are accurate portrayals of actual system and noise dynamics, then the Kalman filter will produce accurate (but not perfect!) state estimates as well as an estimate of uncertainty associated with its state estimates.

Though MMAE theory is not limited to linear, time-invariant applications, it is assumed for this thesis that actual system dynamics and measurements are adequately represented during the time of interest by a reduced-order, linear, time-invariant, discrete-time set of state and measurement equations:

$$\mathbf{x}(t_{i+1}) = \mathbf{\Phi}\,\mathbf{x}(t_i) + \mathbf{B}_d\mathbf{u}(t_i) + \mathbf{w}_d(t_i) \tag{2-1}$$

$$\mathbf{z}(t_i) = \mathbf{H}\,\mathbf{x}(t_i) + \mathbf{D}_z\mathbf{u}(t_i) + \mathbf{v}_d(t_i) \tag{2-2}$$

where $\mathbf{x}$ is the vector of enough system states to describe the system adequately, $\mathbf{u}$ is the vector of control inputs, $\mathbf{z}$ is the vector of actual system measurements, and where dynamics driving noise (uncertainty in the model), $\mathbf{w}_d(t_i)$, and measurement corruption noise, $\mathbf{v}_d(t_i)$, representing uncertainty in the measurement, are both discrete-time, white Gaussian noises with statistics:

$$E\{\mathbf{w}_d(t_i)\} = \mathbf{0} \qquad E\{\mathbf{w}_d(t_i)\,\mathbf{w}_d^T(t_i)\} = \mathbf{Q}_d$$

$$E\{\mathbf{v}_d(t_i)\} = \mathbf{0} \qquad E\{\mathbf{v}(t_i)\,\mathbf{v}^T(t_i)\} = \mathbf{R}$$

$$E\{\mathbf{w}_d(t_i)\,\mathbf{v}^T(t_i)\} = \mathbf{0} \tag{2-3}$$

Between measurement sample times, the k^{th} Kalman filter propagates a state estimate and estimation error covariance (description of uncertainties):

$$\hat{\mathbf{x}}_k(t_{i+1}^-) = \mathbf{\Phi}_k \hat{\mathbf{x}}_k(t_i^+) + \mathbf{B}_{dk}\mathbf{u}_k(t_i) \tag{2-4}$$

$$\mathbf{P}_k(t_{i+1}^-) = \mathbf{\Phi}_k \mathbf{P}_k(t_i^+)\mathbf{\Phi}_k^T + \mathbf{G}_{dk}\mathbf{Q}_{dk}\mathbf{G}_{dk}^T \tag{2-5}$$

starting from known initial conditions, $\hat{\mathbf{x}}_k(t_0)$ and $\mathbf{P}_k(t_0)$. The filter is updated at the next measurement sample time by:

$$\mathbf{A}_k(t_i) = \mathbf{H}_k \mathbf{P}_k(t_i^-)\mathbf{H}_k^T + \mathbf{R}_k \tag{2-6}$$

$$\mathbf{K}_k(t_i) = \mathbf{P}_k(t_i^-)\mathbf{H}_k^T \mathbf{A}_k^{-1}(t_i) \tag{2-7}$$

$$\mathbf{r}_k(t_i) = \mathbf{z}(t_i) - \mathbf{H}_k \hat{\mathbf{x}}_k(t_i^-) - \mathbf{D}_z\mathbf{u}(t_i) \tag{2-8}$$

$$\hat{\mathbf{x}}_k(t_i^+) = \hat{\mathbf{x}}_k(t_i^-) + \mathbf{K}_k(t_i)\,\mathbf{r}_k(t_i) \tag{2-9}$$

$$\mathbf{P}_k(t_i^+) = \mathbf{P}_k(t_i^-) - \mathbf{K}_k(t_i)\mathbf{H}_k \mathbf{P}_k(t_i^-) \tag{2-10}$$

where the - and + superscripts indicate immediately before and after the measurement is taken, respectively, at sample time t_i. It is important to note at this point, that the actual implementation of Equations (2-5), (2-6), (2-7), and (2-10) will be unnecessary, since this thesis will assume that the filters are in steady state operation and, under those assumptions, those equations yield constant values. The subscript $k; 1 \le k \le K$ on the dynamics, propagation, and update equations tells us that this particular Kalman filter is one of K filters, some of which, due to changes in the actual system being modeled, may describe system behavior more accurately than others at different times. For example, given a set of equations for one Kalman filter, we assume, or *hypothesize* that those equations "fit" the behavior of the system and small uncertainties in $\mathbf{\Phi}$, $\mathbf{B}_d$, or $\mathbf{H}$ are modeled stochastically through dynamics and measurement noise. Perhaps we know, however, that actual system parameters might change in some way such that, unless steps are taken to "adapt" to these changes, our Kalman filter state equations will no longer be adequate portrayals of system behavior and

inaccurate estimates will result. We don't know exactly "when" or "how much" the parameters change, but do know "within what bounds" with reasonable certainty. Since we cannot design an infinite number of Kalman filters to cover all possibilities, we "discretize the parameter space" by designing "K" discrete Kalman filters, each based on an important hypothesis or model of system dynamics, and blend their estimates optimally to obtain a final state estimate and, if needed, an estimate of system parameters. This is the concept behind MMAE: a bank of K Kalman filters, each filter based on different assumed models of system dynamics, measurements, and/or statistics. The state and parameter estimates corresponding to K assumed models are blended to obtain system state and parameter estimates. Now that we understand the concept, the next section describes how the actual MMAE algorithm is constructed.

2.6. MMAE Algorithm.

(For ease of explanation, the MMAE algorithm will be developed for hypotheses of *total* actuator and sensor impairments, with the understanding that an extension, or "growth", of the algorithm is necessary for the study of dual *partial* impairments, which will be developed later in Chapter 4). Let **a** be a vector of uncertain parameters in a linear stochastic state model for a dynamic system. In this thesis application, **a** represents all possible VISTA F-16 total failure modes. Since there are an infinite number of impairments (when considering partial ones) possible, the parameter space must be discretized, preferably to as small a number, K, of failure modes $\mathbf{a}_k$, k = 1,2...K, as possible. For this thesis, the parameter space (for *total* failures) is composed of the fully functional VISTA F-16 at Mach .4, 20000 feet, five individual total actuator impairments, six individual total sensor impairments, and all possible actuator and sensor total im-

MMAE Channel Abbreviation	Fully Failed Filter Hypothesis (Actuators)	MMAE Channel Abbreviation	Fully Failed Filter Hypothesis (Sensors)
FF	Fully Functional (No Failure)	AOA	Angle-of-Attack
LS	Left Stabilator	Q	Pitch Rate
RS	Right Stabilator	A_n	Normal Acceleration
LF	Left Flaperon	P	Roll Rate
RF	Right Flaperon	R	Yaw Rate
RUD	Rudder	A_y	Lateral Acceleration

Table 2-1 A List of Actuator and Sensor Total Impairments Modeled in the MMAE Algorithm

pairment combinations of the total impairments listed in Table 2-1.

The number of possible "total actuator impairment" modes is thus: $1+K+\frac{K!}{(K-2)!2!}=67$, plus an infinite number of partial actuator impairments which are to be estimated by a weighted average of outputs from the Kalman filters corresponding to totally failed and fully functional hypotheses. One should ask at this point if the level of parameter discretization is too coarse, since there are only "fully functional" and "fully failed" hypotheses being modeled. The answer to this question is that there are *no* guarantees that we have discretized the failure space to an adequate degree for this thesis, but since prior research efforts [14, 26, 40, 45] obtained good results using this level of discretization, we elect to proceed. It is understood that additional Kalman filters may (and will) have to be designed and implemented at a later time.

We define next a hypothesis conditional probability, $p_k(t_i)$, as the "probability that **a** assumes the value $\mathbf{a}_k$ (for k = 1,2...K: possible discrete values), conditioned on the observed history of measurements up to time t_i":

$$p_k(t_i) = \text{prob}\{\mathbf{a}=\mathbf{a}_k \mid \mathbf{Z}(t_i)=\mathbf{Z}_i\} \tag{2-11}$$

where $\mathbf{Z}(t_i)$ is the history of all measurements from $\mathbf{z}(t_1)$to $\mathbf{z}(t_i)$. It can be shown [27, 35] that $p_k(t_i)$ can be evaluated recursively for all k via:

$$p_k(t_i) = \frac{f_{\mathbf{z}(t_i)|\mathbf{a},\mathbf{Z}(t_{i-1})}(\mathbf{z}_i \mid \mathbf{a}_k, \mathbf{Z}_{i-1})\, p_k(t_{i-1})}{\sum_{j=1}^{K} f_{\mathbf{z}(t_i)|\mathbf{a},\mathbf{Z}(t_{i-1})}(\mathbf{z}_i \mid \mathbf{a}_j, \mathbf{Z}_{i-1})\, p_j(t_{i-1})} \tag{2-12}$$

We notice that the current probability value is based on the last value, $p_k(t_{i-1})$, weighted by the density function describing "the incoming measurement **z** at time t_i, given an assumed value of parameter $\mathbf{a}_k$ and an observed value of the previous measurement history $\mathbf{Z}(t_{i-1})$", and normalized by the sum of all such values corresponding to each point in the parameter space. Hence, the resulting probabilities must behave in the following manner:

$$p_k(t_i) \geq 0$$

$$\sum_{k=1}^{K} p_k(t_i) = 1 \tag{2-13}$$

(Note that, when we say the "probability that **a** assumes the value $\mathbf{a}_k$..." in this thesis, we mean: "the probability that the failure status, **a**, is best described by failure condition (or hypothesis) k, based on the history of sensor measurements".) The Gaussian conditional density function given in Equation (2-12) is the density function for the current measurement, **z**, based on the assumed parameter value (the failure hypothesis), $\mathbf{a}_k$, and the entire history of previous measurements:

$$f_{\mathbf{z}(t_i)|\mathbf{a},\mathbf{Z}(t_{i-1})}(\mathbf{z}_i \mid \mathbf{a}_k, \mathbf{Z}_{i-1}) = \beta_k \exp\{\cdot\} \tag{2-14}$$

where:

$$\{\cdot\} = \{-\tfrac{1}{2}\mathbf{r}_k^T(t_i)\mathbf{A}_k^{-1}(t_i)\mathbf{r}_k(t_i)\} \tag{2-15}$$

and:

$$\beta_k = \frac{1}{(2\pi)^{m/2}\,|\mathbf{A}_k(t_i)|^{1/2}} \tag{2-16}$$

Note that this is also the density function for the residuals of the k^{th} Kalman filter, which are white, Gaussian and have zero mean and covariance $\mathbf{A}_k$. The "likelihood quotient" for the k^{th} Kalman filter, L_k, is defined as follows:

$$L_k(t_i) = \mathbf{r}_k^T(t_i)\mathbf{A}_k^{-1}(t_i)\mathbf{r}_k(t_i) \tag{2-17}$$

If this filter's hypothesis (its internal model) matches the true failure mode of the system, then its residuals $\mathbf{r}_k$ will have a mean squared value agreeing closely with the corresponding residual covariance predicted by the same filter, the likelihood quotient will approach m (the dimension of $\mathbf{r}_k$, and $\mathbf{z}_k(t_i)$), and $p_k(t_i)$ will increase. If this filter's hypothesis does not match the true system failure, then actual

measurements will differ widely from the filter-predicted values, residuals and likelihood quotient will grow in magnitude, and $p_k(t_i)$ will decrease.

Now we have a bank of K separate Kalman filters, each based on a particular value $\mathbf{a}_1$, $\mathbf{a}_2$, ..$\mathbf{a}_k$...$\mathbf{a}_K$ of the parameter vector, and corresponding probabilities, or "weights" that the particular hypothesis is correct: $p_1(t_i)$, $p_2(t_i)$, ... $p_k(t_i)$,..$p_K(t_i)$. While we could choose to take the state estimate associated with the highest value of probability and call that the best estimate, called a maximum a posteriori (MAP) estimate, we choose instead a Bayesian estimation approach which forms a conditional mean rather than a conditional mode, and blends estimates from *all* filters:

$$\hat{\mathbf{x}}_{MMAE}(t_i^+) = \mathrm{E}\{\mathbf{x}(t_i) \mid \mathbf{Z}(t_i) = \mathbf{Z}_i\} = \sum_{k=1}^{K} \hat{\mathbf{x}}_k(t_i^+)\, p_k(t_i) \tag{2-18}$$

where $\hat{\mathbf{x}}_k(t_i^+)$ is the state estimate generated by Kalman filter k, which is based on the parameter value corresponding to $\mathbf{a}_k$. In choosing this Bayesian estimation approach, it is possible to blend estimates from adjacent filters when the true system failure is between discrete hypotheses. Since this thesis studies partial losses of actuator effectiveness, we need an estimation method with these attributes.

Though not required for the online implementation of MMAE studied in this thesis, it is a simple task, if needed, to generate error covariances for state and/or parameter estimates [35]. Furthermore, if an estimate of the true failure hypothesis is required:

$$\hat{\mathbf{a}}_{MMAE}(t_i^+) = \mathrm{E}\{\mathbf{a}(t_i) \mid \mathbf{Z}(t_i) = \mathbf{Z}_i\} = \sum_{k=1}^{K} \mathbf{a}_k\, p_k(t_i) \tag{2-19}$$

A simulation representation of the MMAE algorithm just described is illustrated in Figure 2-1.

All Kalman filters in the parallel bank are given the same sensor measurements $\mathbf{z}_i$ and control inputs $\mathbf{u}$. Each filter uses the input information to produce its prediction of system behavior, one time step into the future, according to Equation (2-4). The filter residuals are formed, using that prediction and the

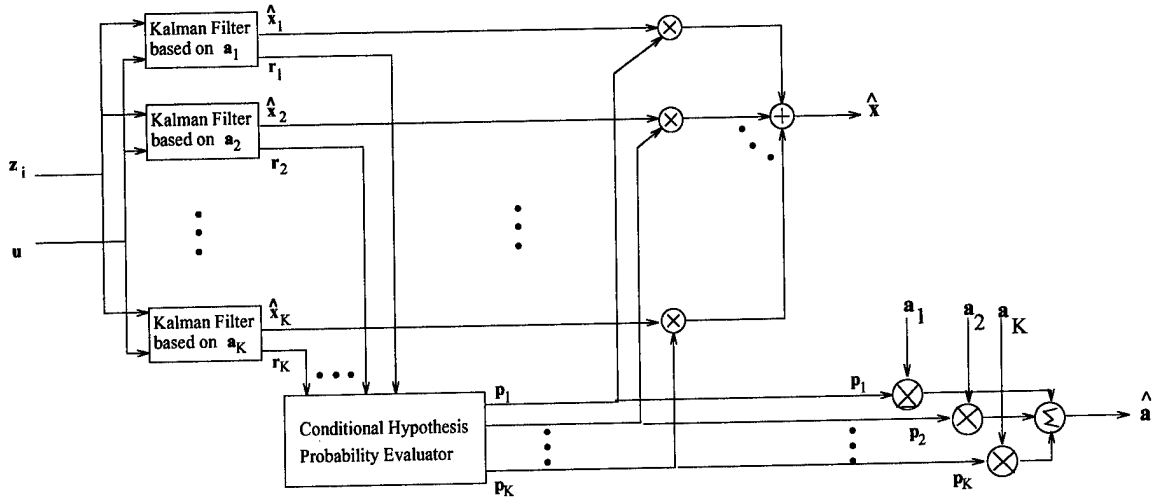
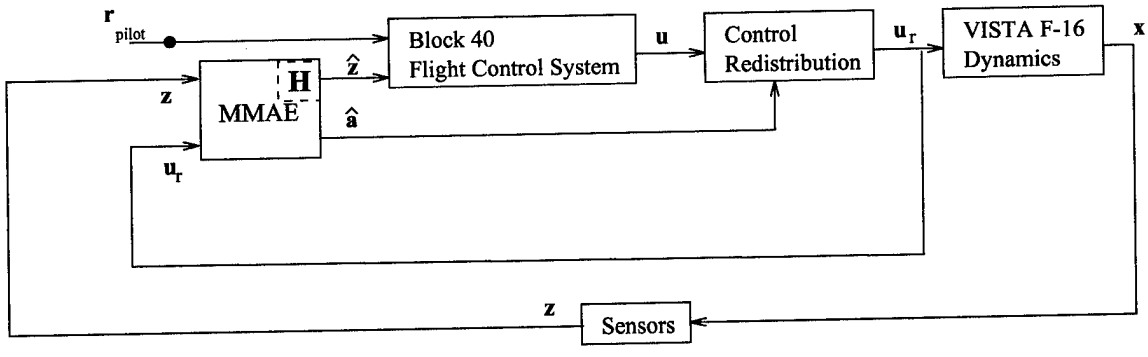


Figure 2-1 MMAE Simulation Diagram

next sensor measurement when it arrives (Equation (2-8)), and are sent to the Conditional Hypothesis Probability Evaluator module, where probabilities for all filters are evaluated according to Equation (2-12). Each filter's updated estimate, $\hat{\mathbf{x}}_k(t_i^+)$, and its assumed hypothesis, $\mathbf{a}_k$, is multiplied by its corresponding conditional probability and summed according to Equations (2-18) and (2-19) to yield final state and parameter estimates.

2.7. *MMAE Interface with Other Modules.*



(Note: MMAE includes measurement model, H)

Figure 2-2 MMAE Interface with Adjacent Modules

Though exact details will be explained later, the MMAE algorithm interfaces with the VISTA F-16 flight control system as shown in Figure 2-2. MMAE replaces the suite of raw sensor measurements, $\mathbf{z}(t_i)$, normally supplied to the VISTA Block 40 flight control system with reconstructed measurements, $\hat{\mathbf{z}}(t_i) = \mathbf{H}\,\hat{\mathbf{x}}_{MMAE}(t_i^-) + \mathbf{D}_z\mathbf{u}_r(t_i)$. Probability estimates, $p_k(t_i)$, and/or parameter estimates $\hat{\mathbf{a}}_{MMAE}$, may be supplied directly to the Control Redistribution module if they are of sufficient quality, or they may be processed further before being used for Control Reconfiguration purposes.

Under normal operating conditions with no failures of either sensors or actuators, the state estimate vector as output by MMAE, $\hat{\mathbf{x}}_{MMAE}(t_i^+)$, allows optimal reconstruction of sensor measurements (as though there were no failed sensors) needed by the VISTA flight control system. With no failures in the real world, the conditional probability estimates computed by MMAE should correspond to a fully functional aircraft, and normal Block 40 flight control system commands would be passed directly to the actuators.

2.8. *Modifications to the MMAE Algorithm*

As stated in the historical buildup of MMAE/MMAC, modifications have been made in past research to enhance detection performance of the algorithm. Algorithm modifications used in this thesis and discussed here include lower bounding and rescaling conditional probabilities, beta dominance compensation, scalar penalty adjustment, and control surface dithering. The hierarchical filter bank structure, used for multiple failure detection, is an *extension* of MMAE/MMAC and is discussed last.

2.8.1. *Lower Bounding and Rescaling.*

As seen in Equation (2-12), the conditional probabilities, $p_k(t_i)$, satisfy recursive calculations, implying that if a probability calculation is ever zero, then it will "latch" and remain at zero for all time afterwards, locking out the corresponding hypothesis and hence the algorithm's sensitivity to that failure. If a probability value is extremely small, then algorithm performance might be sluggish as probability values

take time to accumulate. To prevent "lock out" and to improve response, a lower bound on p_k equal to .001 was introduced in prior research and has been used with success since [14, 26, 31, 33, 40, 45]. Although a lower bound on p_k prevents lock out, it also now admits small non-zero components into state and parameter computations (Equations (2-18) and (2-19)) which correspond to totally erroneous hypotheses. To remedy that problem, a second, lower bound was introduced by Eide [14] to filter out probabilities with values less than .003. In both cases of artificial lower bounds, the probabilities were rescaled to ensure summation to one.

2.8.2. *Beta Dominance Compensation.*

It is possible, within the MMAE algorithm, for all Kalman filters in the bank eventually to have L_k's which are approximately equal. If that is the case, then the exponential components of all K filters have the same approximate value, leaving the β_k terms in front to dominate according to their computed values of $|\mathbf{A}_k(t_i)|$ (see Equation (2-6)). A small, or relatively small, $|\mathbf{A}_k(t_i)|$ value yields a large value of β_k and a large value of $p_k(t_i)$. If we recall that $\mathbf{A}_k(t_i) = \mathbf{H}_k\mathbf{P}_k(t_i^-)\mathbf{H}_k^T + \mathbf{R}_k$, then it becomes clear that a Kalman filter within an MMAE bank and modeling a sensor failure hypothesis with a zero row of $\mathbf{H}_k(t_i)$, will have a much smaller $\mathbf{A}_k$ than will a filter modeling an actuator failure hypothesis. The result, borne out in previous research [1], will be a tendency towards declaring sensor failure false alarms. Since Equation (2-12) is normalized by the sum of the numerator terms of all filters, and since all useful information for failure declaration is contained in the residuals, $\mathbf{r}_k$, it was proposed and demonstrated [30, 46] that the leading term, $\beta_k = \frac{1}{(2\pi)^{m/2}|\mathbf{A}_k(t_i)|^{1/2}}$, could be stripped from all calculations, reducing or eliminating the predisposition towards declaring sensor failures.

2.8.3. *Scalar Penalty Adjustment*

The scalar coefficient of '$-\frac{1}{2}$' in exponential terms '$\exp\{-\frac{1}{2}\mathbf{r}_k^T(t_i)\mathbf{A}_k^{-1}(t_i)\mathbf{r}_k(t_i)\}$' can be used as a tuning parameter, if needed, to increase sensitivity of p_k to large residuals [31]. Care must be taken

however, since increasing this "scalar penalty" also increases sensitivity and incidence of false alarms. This research will not use this option, and will leave the value equal to $-\frac{1}{2}$.

2.8.4. *Dithering.*

Dithering can be described as an introduction of low magnitude control, or "probing" signals with the intent of exciting the system and enhancing identifiability. It is often necessary, especially in a simulation environment, to "shake the system up" with probing signals in order to get residual activity substantial enough to identify and differentiate between possible failure hypotheses. Past research has demonstrated [11, 26, 45], after experiments with many different waveforms of different amplitudes and frequencies, that a sinusoid signal of 15 radians/second appears to be the most effective signal for the application studied in this thesis. To be subliminal to the pilot, the dither command to the VISTA F-16 control system is kept small enough so that resulting aircraft accelerations are limited to between ± .1g's longitudinally and ± .2g's laterally. To counter the (intentional) coupling effects of the F-16 Aileron to Rudder Interconnect (ARI), the dither signal to the longitudinal and lateral/directional channels of the flight control system are 90° out of phase [45]. Although dither may not be necessary in an actual, maneuvering and turbulent flight environment, and is certainly not desirable at all on a continuous basis, it is necessary to this research, since *all* test cases performed in Chapter 4 are under conditions of no aircraft maneuvers.

2.8.5. *Hierarchical Filter Bank Structure.*

The primary goal of this thesis is to study Multiple Model Adaptive Estimation with Control Redistribution (MMAE/CR) performance, during total and partial, single and dual control surface impairments under non-maneuvering conditions. The VISTA F-16 truth model, as simulated on the SRF and used in this research, models five actuators and six sensors. Thus an MMAE bank of "single failure hypotheses", $\mathbf{a}_{k'}$, has N=11 Kalman filters, each modeling a single failure, plus one more filter modeling a

"fully functional aircraft" (ff) hypothesis, for a total of N+1=12 filters in parallel. When all possible hypotheses of dual, total actuator impairments are added to the single ones, the number of Kalman filters in parallel grows to: $1+N+\frac{N!}{(N-2)!2!}=67$. This is clearly not a feasible MMAE algorithm since computational speed and algorithm complexity must be minimized in onboard flight control computer applications. The solution, developed and tested with success by Stevens and others [14, 26, 45] , is to implement the hierarchical filter structure shown in Figure 2-3:

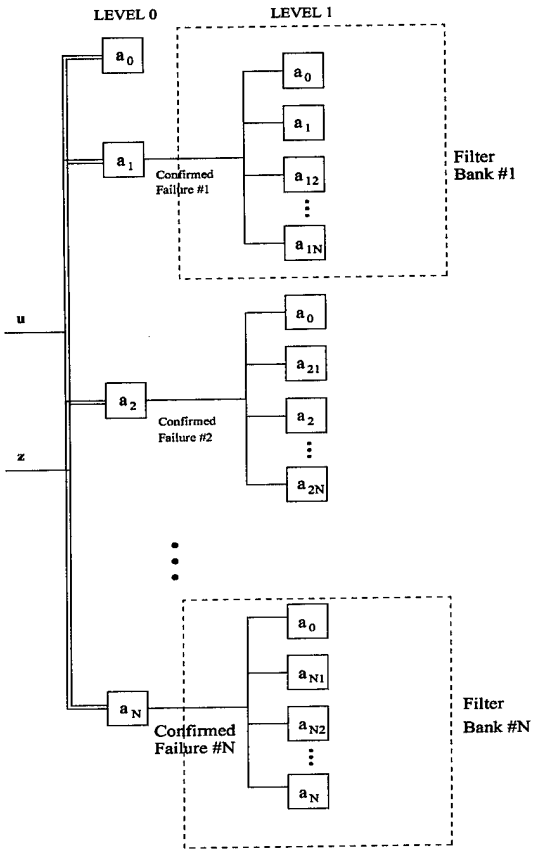


Figure 2-3 Hierarchical MMAE Filter Structure for Dual Failures

Failure hypotheses are grouped in banks of twelve (N+1), and only one bank is allowed to be "on line" at any time. If a failure is detected by any of the eleven parallel, single-failure filters composing "Level 0", then the "Level 0" bank is switched out and replaced with the "Level 1" bank which hypothesizes the first total failure plus any second total failure, plus a filter hypothesizing the first failure only (no second failure). Furthermore, since it is possible that a false declaration is made of the first failure, each "Level 1" bank also includes a filter based on a fully functional aircraft hypothesis, allowing the MMAE algorithm to reverse its

decision , back out of the "Level 1" bank, and return to "Level 0". The reader should be advised that Figure 2-3 does not show expansions of the MMAE algorithm, in the form of added filter banks, that are necessary for detection of dual impairment scenarios containing a partial first impairment. Those are discussed in detail in Chapter 4.

2.9. Chapter Summary

This chapter began with a history of MMAE/MMAC research preceding this thesis effort. The MMAE algorithm was constructed in detail, starting with its basic component, the Kalman filter. The conditional probability calculations , used to weight the Kalman filter estimates, were discussed next, and the resulting weighted estimates were combined to yield a blended, or Bayesian estimate of the total state. The interface between the MMAE algorithm and the VISTA flight control system was discussed, followed by modifications and extensions which have been made to enhance MMAE performance. The modifications used in this thesis are: lower bounding and rescaling, stripping the β term from probability calculations, and the use of dither commands to enhance detection and identifiability, and using the hierarchical filter structure for multiple impairments. The next chapter will focus on specifics of the SRF/VISTA F-16 truth model, the method of inserting impairments, and the resulting linear models used for MMAE Kalman filter design. The history and implementation of Control Redistribution (CR) will be discussed in detail, as well as the interface between MMAE and CR necessary for this research.

3. *Truth Model, MMAE Model, and Control Redistribution: Development*

3.1. Chapter Overview

This chapter begins with a short history of the simulation environment used in this thesis: the Simulation Rapid Prototyping Facility, or SRF. An introduction to the VISTA F-16 aircraft is given next, followed by a description of the non-realtime simulation of that aircraft on the SRF, which serves as the "truth model" for this thesis research. The method of simulating total actuator failures and partial losses of control surface effectiveness, or "soft failures", within the truth model is also discussed.

The chapter then focuses on construction of linear, reduced order aircraft, actuator, and measurement models used by the Kalman Filters in the MMAE algorithm, as well as the failure modeling method used. Next, the Control Redistribution (CR) concept is explained, including history, algorithm development, and its interface with MMAE and the VISTA flight control system. The chapter ends with a summary.

3.2. A Short History of the Simulation Rapid Prototyping Facility (SRF)

The SRF came in to existence in the mid-1980's due to the Air Force's "Reliability and Maintainability 2000" program, one thrust of which was the study of Reconfigurable Flight Control Systems undertaken at Wright Labs [7, 8, 11, 12]. To support this thrust, the SRF was developed [12, 23], originally on VAX computers, to be a user-friendly environment on which to host various non-realtime aircraft simulations for research. Given a promising reconfigurable flight control algorithm (from a contractor, or from "in house"), the concept was to implement, study, and optimize performance on the SRF-based simulation, then move to real-time computers for man-in-loop testing and perhaps a flight test. Two simulations of note were the HIDEC F-15, which made it to flight test, and the Control Reconfigurable Combat Aircraft (CRCA) simulation [7, 12]. With the start of VISTA F-16 aircraft construction, the Air

Force in 1989 acquired the VISTA simulation from General Dynamics [12], and the SRF became part of the "VISTA Vision" concept [12]. VISTA Vision was an entire "initial concept-to-flight test design track" for flight control that started with proof-of-concept evaluations on the SRF, progressed to real-time man-in-loop simulation, and ended with a flight test aboard the VISTA aircraft. Though the VISTA Vision concept has since been abandoned, the SRF remains as a simulation environment for "proof-of-concept" flight control research.

3.3. VISTA F-16

The Variable In-flight Stability Test Aircraft (Figure 3.1) gives the Air Force a capability to simulate existing or future high-performance aircraft in actual flight. By following a desired model and flight control system resident on the Variable Stability System, or VSS, the VISTA F-16 can be made to feel like a completely different aircraft, and, unlike any earth-bound simulator, provides perfect motion cues to the pilot.

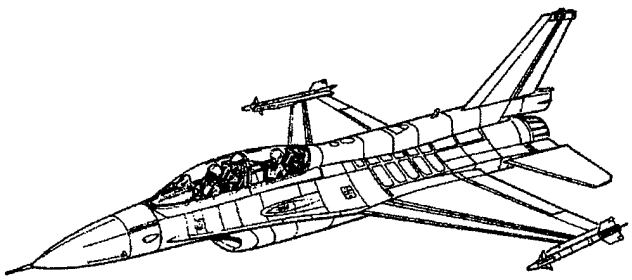


Figure 3-1 ***The VISTA F-16***

For reasons described in Section 3.2, the complete VISTA simulation presently on the SRF was acquired directly from General Dynamics in 1989. The SRF/VISTA simulation was used to research questions about possible switching transients between "VSS" and "normal F-16" modes, and to prepare for real-time simulation testing. During real-time, man-in-loop simulation testing, the flight characteristics of the VISTA F-16 simulation *in the normal, subsonic, F-16 flight envelope* were confirmed by actual F-16 pilots [12]. This fact, plus the extensive simulation data available, makes the SRF/VISTA F-16 simulation an excellent "truth model" to use in MMAE/CR or MMAC research.

VISTA F-16 Simulation Truth Model Description

When the VISTA F-16 simulation was first delivered, it was the highest fidelity simulation available. Undoubtedly, there have been many changes made on GD's (now Lockheed's) simulation since then which haven't been implemented on the SRF version. Regardless, the SRF version is still a highly accurate "truth model" for MMAE/CR research. It is a fully non-linear, full-envelope, six-degree-of-freedom simulation, which includes Fortran77 representations of the Block 40 Flight Control System with Aileron-Rudder Interconnect (ARI), fourth-order actuators with surface position and rate limiting, and equations of motion which employ user-adjustable mass properties (weight, inertias, c.g. location) and accurate F-16 aerodynamic data to build up total forces and moments. The engine is modeled using a first order lag with coefficients scheduled as functions of Mach and altitude. The VSS, described earlier, is not used in this thesis research. The simulation did not include a turbulence model when delivered originally, so one was added later to the SRF, and improved upon in later research [14].

3.4. MMAE Integration Modifications.

Modifications "on top of" the existing SRF simulation have been made during prior research efforts to accommodate the integration and research of MMAC and MMAE-based control. In addition to the improved turbulence model, sensor noise was added, normal and lateral acceleration sensor models

were added [14], and software was added to allow simulation of simultaneous dual failures [14]. Lewis [26] reactivated the hierarchical form of MMAE with "bank swapping" (Section 2.7.5) for his study of multiple failures.

3.5. *Truth Model Failure Simulation and Insertion.*

The primary focus of this thesis is to study MMAE/CR performance under *partial* effectiveness impairments (losses) of single and dual combinations of control surfaces. All previous research efforts using the SRF/VISTA F-16 simulation [14, 26, 45] studied "hard failures" of single or dual surfaces by commanding the affected surfaces on the truth model to their [non-zero] trim positions (measured with respect to the aircraft), which are equivalent to zero perturbation (measured with respect to trim) positions. Although this method of modeling an actuator failure does *not* approximate a failure to a free stream position (which approximately equals and changes with angle of attack so that the surface exerts no force or moment on the aircraft) position, it is a good choice which guarantees agreement between the truth model and the linear model, as well as a reasonably accurate portrayal of an actuator failure on an F-16, which fails to zero degrees deflection. Previous research projects could legitimately call their failures "total actuator failures" and were correct.

In this thesis, we simulate a partial actuator failure, or "impairment" in the F-16 truth model (specifically, in the actuators) in a linear fashion, multiplying by an "effectiveness scaling factor", ε: ; $0 \le \varepsilon \le 1$, effectively to blend between the total actuator deflection (for no impairment, or $\varepsilon = 1$) and the trim deflection (for a total impairment, or $\varepsilon = 0$):

$$\delta_{act} = \varepsilon\delta_{act} + (1-\varepsilon)\delta_{trim} \tag{3-1}$$

Strictly speaking, we are not really modeling partial, or "soft", actuator failures since that would likely entail non-linear effects like reduced rate and position limits. We are also not modeling aerodynamic changes, which would probably result from the damage caused by a partial surface loss. This impairment

model was chosen because it converges to prior results "at the endpoints", namely, for $\varepsilon = 0$, the model reproduces Stepaniak's fully failed actuator model, and for $\varepsilon = 1$, the model outputs normal actuator commands.

3.6. *Reduced Order, Linearized Model for MMAE Kalman Filter Design*

As explained in Chapter 2, each Kalman Filter in the MMAE algorithm is based upon a reduced order representation of the truth model, a *design model*, of adequate fidelity to yield desired filter performance. Although all design models for the MMAE algorithm used in this thesis have been developed and explained well in prior research efforts [14, 45], this section will re-examine their derivation. The general steps to be followed in order are: 1) obtaining an adequate linear, time invariant (LTI) state space VISTA F-16 model, yielding continuous-time, open loop **A, B, C, D** matrices, 2) expanding the **B** matrix from total elevator and aileron deflection to single-surface deflection contributions, 3) selecting an adequate reduced order actuator model and appending it to the existing state space equations, 4) implementing a method of inserting impairments into the Kalman filters (versus into the truth model), 5) formulating discrete-time measurement and output models, to include acceleration sensors, 6) adding continuous-time plant dynamics noise and discrete-time measurement noise, and, 7) discretizing the continuous-time equations to obtain the final equivalent [34] discrete-time Kalman filter design model that was assumed in Chapter 2.

3.7. *Plant Matrices*

For the MMAE algorithm used in this research, the Kalman Filters are based on a linear, time invariant, discrete-time state space model of the VISTA F-16 at flight condition: Mach=.4, altitude=20000 feet. We start, however, with the continuous-time model:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{3-2}$$

where the full longitudinal and lateral-directional, dimensional, stability and control matrices are defined as:

$$\mathbf{A}=\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ X'_\theta & X'_u & X'_\alpha & X'_q & 0 & 0 & 0 & 0 \\ Z'_\theta & Z'_u & Z'_\alpha & Z'_q & 0 & 0 & 0 & 0 \\ M'_\theta & M'_u & M'_\alpha & M'_q & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \phi'_r \\ 0 & 0 & 0 & 0 & Y'_\phi & Y'_\beta & Y'_p & Y'_r \\ 0 & 0 & 0 & 0 & 0 & L'_\beta & L'_p & L'_r \\ 0 & 0 & 0 & 0 & 0 & N'_\beta & N'_p & N'_r \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ X'_{\delta_e} & 0 & X'_{\delta_f} & 0 & 0 \\ Z'_{\delta_e} & 0 & Z'_{\delta_f} & 0 & 0 \\ M'_{\delta_e} & 0 & M'_{\delta_f} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & Y'_{\delta_{dt}} & 0 & Y'_{\delta_a} & Y'_{\delta_r} \\ 0 & L'_{\delta_{dt}} & 0 & L'_{\delta_a} & L'_{\delta_r} \\ 0 & N'_{\delta_{dt}} & 0 & N'_{\delta_a} & N'_{\delta_r} \end{bmatrix} \tag{3-3}$$

x	State Variables	Units
1	θ Pitch Angle	rad
2	u Forward velocity	ft/sec
3	α Angle of attack	rad
4	q Pitch rate	rad/sec
5	ϕ Bank angle	rad
6	β Sideslip angle	rad
7	p Roll rate	rad/sec
8	r Yaw rate	rad/sec
u	**Control Inputs** (Perturbations)	**Units**
1	δ_e Elevator position	rad
2	δ_{dt} Differential tail position	rad
3	δ_f Flap position	rad
4	δ_a Aileron position	rad
5	δ_r Rudder position	rad

Table 3-1 State and Control Input Vectors for Equation (3-2)

and the state and control input vectors are defined as in Table 3-1. At this flight condition (Mach .4, 20000 feet, standard day), with the aircraft mass properties defined by selecting the configuration to be "up and away" with 2 AIM 9L's, and with wing tanks empty, the SRF yields the following values for the body axes stability and control derivatives [14, 26, 45]:

$X'_\theta = -31.6771\ (ft/sec^2)$	$Z'_\theta = -.141578E\text{-}01\ (1/sec)$	$M'_\theta = 0.642121E\text{-}03\ (1/sec)$
$X'_u = 0.243115E\text{-}02\ (1/sec)$	$Z'_u = -0.195975E\text{-}03\ (1/ft)$	$M'_u = -0.134645E\text{-}02\ (1/ft\text{-}sec)$
$X'_\alpha = 16.3530\ (ft/sec^2)$	$Z'_\alpha = -0.440414\ (1/sec)$	$M'_\alpha = 1.52512\ (1/sec^2)$
$X'_q = -73.4589\ (ft/sec)$	$Z'_q = 0.997196$	$M'_q = -0.526916\ (1/sec)$
$X'_{\delta_e} = 2.08784\ (ft/sec^2)$	$Z'_{\delta_e} = -0.684394E\text{-}01\ (1/sec)$	$M'_{\delta_e} = -3.64478\ (1/sec^2)$
$X'_{\delta_f} = -0.542584\ (ft/sec^2)$	$Z'_{\delta_f} = -0.337867E\text{-}01\ (1/sec)$	$M'_{\delta_f} = 0.285675\ (1/sec^2)$
$Y'_\phi = 0.775991\ (1/sec)$		$\phi'_r = 0.182448$
$Y'_\beta = -0.109880\ (1/sec)$	$L'_\beta = -18.5276\ (1/sec^2)$	$N'_\beta = 2.83301\ (1/sec^2)$
$Y'_p = 0.180844$	$L'_p = -1.55588\ (1/sec)$	$N'_p = -0.432911E\text{-}01\ (1/sec)$
$Y'_r = -0.997627$	$L'_r = 0.413477E\text{-}01\ (1/sec)$	$N'_r = -0.282169\ (1/sec)$
$Y'_{\delta_{dt}} = 0.13771E\text{-}01\ (1/sec)$	$L'_{\delta_{dt}} = -8.99043\ (1/sec^2)$	$N'_{\delta_{dt}} = -1.03452\ (1/sec^2)$
$Y'_{\delta_a} = 0.569506E\text{-}03\ (1/sec)$	$L'_{\delta_a} = -12.4072\ (1/sec^2)$	$N'_{\delta_a} = -0.130713\ (1/sec^2)$
$Y'_{\delta_r} = 0.169586E\text{-}01\ (1/sec)$	$L'_{\delta_r} = 2.86294\ (1/sec^2)$	$N'_{\delta_r} = -1.16519\ (1/sec^2)$

Table 3-2 Primed Dimensional Stability and Control Derivatives

resulting in the following eigenvalues for the open-loop, or "bare airframe" plant:

Longitudinal Eigenvalues:	Lateral-directional Eigenvalues:
$p_1 = -0.0268 + 0.1376j$	$p_1 = -0.3365 + 2.4071j$
$p_2 = -0.0268 - 0.1376j$	$p_2 = -0.3365 - 2.4071j$
$p_3 = 0.8341$ **(unstable)**	$p_3 = -0.0444$
$p_4 = -1.7455$	$p_4 = -1.2305$

Table 3-3 F-16 Open-loop Plant Eigenvalues

For an explanation of body axes, and stability and control derivatives, the interested reader should refer to Blakelock [5] and Nelson [43]. Note, however, that plant matrix **A** contains an unstable longitudinal eigenvalue, which makes the flight control problem more challenging. Partial loss of control surface effectiveness will mean some loss of augmented longitudinal stability as well, so MMAE detection speed and Control Redistribution performance will be of crucial importance. Stepaniak [45] also showed that the aircraft is not in perfect trim, but since its flight path deviates only slightly over eight seconds, this is not a great concern.

3.7.1. *Modified B Matrix.*

The input vector **u** in Table 3-1 gives total perturbation deflections, in degrees, of the elevators, ailerons, differential tails (the elevators deflected asymmetrically for roll), flaps and rudder. Leading edge flaps are not commanded by the linear model, because they are deflected automatically as a function of angle of attack, even during linearization of the open loop airframe. We need to modify the control matrix, **B**, to be compatible with individual surface deflections, since we will be inducing impairments and also because Control Redistribution will be modifying the conventional elevator and aileron relationships. Since [14]:

$$\begin{bmatrix} \delta_e \\ \delta_{dt} \\ \delta_f \\ \delta_a \\ \delta_r \end{bmatrix} = \begin{bmatrix} .5 & .5 & 0 & 0 & 0 \\ -.5 & .5 & 0 & 0 & 0 \\ 0 & 0 & .5 & .5 & 0 \\ 0 & 0 & -.5 & .5 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \delta_{ls} \\ \delta_{rs} \\ \delta_{lf} \\ \delta_{rf} \\ \delta_r \end{bmatrix} \tag{3-4}$$

we can write:

$$\mathbf{B_{mod}} = \mathbf{B} \cdot \begin{bmatrix} .5 & .5 & 0 & 0 & 0 \\ -.5 & .5 & 0 & 0 & 0 \\ 0 & 0 & .5 & .5 & 0 \\ 0 & 0 & -.5 & .5 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3-5}$$

Performing the multiplication and inserting the values from Table 3-2, we get:

$$\mathbf{B_{mod}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1.0667 & 1.0667 & -0.1625 & -0.1625 & 0 \\ -0.0343 & -0.0343 & -0.0102 & -0.0102 & 0 \\ -1.8416 & -1.8416 & 0.0902 & 0.0902 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -0.0069 & 0.0069 & -0.0003 & -0.0003 & 0.0170 \\ 4.5683 & -4.5683 & 6.2649 & -6.2649 & 2.8853 \\ 0.5318 & -0.5318 & 0.0686 & -0.0686 & -1.1784 \end{bmatrix} \tag{3-6}$$

with the modified input vector, $\mathbf{u}_{mod}$, given in Table 3-4:

$\mathbf{u}_{mod}$	Modified Control Inputs	Units
1	δ_{ls} Elevator position	rad
2	δ_{rs} Differential tail position	rad
3	δ_{lf} Flap position	rad
4	δ_{rf} Aileron position	rad
5	δ_{rud} Rudder position	rad

Table 3-4 Modified Input Vector Components

the linear representation of the F-16 (at Mach = .4, 20000 ft.) is now:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}_{mod}\mathbf{u}_{mod} \tag{3-7}$$

3.7.2. Reduced Order Actuator Model.

The next step in model development is to obtain simplified but adequate models of the actuators, and append them to the basic model developed thus far. In the SRF simulation, all five control surface actuators are modeled with the fourth order transfer function:

$$\frac{\delta_{act}}{\delta_{com}} = \frac{(20.2)(144.8)(71.4)^2}{(s+20.2)(s+144.8)(s^2+2(0.736)(71.4)s+71.4^2)} \tag{3-8}$$

where δ_{act} is the actuator position and δ_{com} is the actuator command. For our Kalman Filter design model, we wish to substitute a much simpler first order actuator model. Classical theory would predict: $\frac{\delta_{act}}{\delta_{com}} = \frac{20.2}{s+20.2}$ as the first order model, but Eide [14] suggested that a 14 radian/second model ($\frac{\delta_{act}}{\delta_{com}} = \frac{14}{s+14}$) seemed to fit actual SRF simulation data best. While this should be an "eyebrow raiser" (at this low dynamic pressure with low surface hinge moments, F-16 actuators should be capable of peak

performance) this thesis will use the 14 radian/second first order lag to compare with both the SRF truth model and all preceding research efforts. The first order actuator model in state space form is:

$$\dot{\delta}_{act} = -14\delta_{act} + 14\delta_{com} \tag{3-9}$$

and the state space representation for five control surfaces is:

$$\dot{\boldsymbol{\delta}}_{\mathbf{act}} = -14\mathbf{I}_{5x5}\boldsymbol{\delta}_{\mathbf{act}} + 14\mathbf{I}_{5x5}\boldsymbol{\delta}_{\mathbf{com}} \tag{3-10}$$

$$\begin{bmatrix} \dot{\delta}_{ls} \\ \dot{\delta}_{rs} \\ \dot{\delta}_{lf} \\ \dot{\delta}_{rf} \\ \dot{\delta}_{rud} \end{bmatrix} = \begin{bmatrix} -14 & 0 & 0 & 0 & 0 \\ 0 & -14 & 0 & 0 & 0 \\ 0 & 0 & -14 & 0 & 0 \\ 0 & 0 & 0 & -14 & 0 \\ 0 & 0 & 0 & 0 & -14 \end{bmatrix} \begin{bmatrix} \delta_{ls} \\ \delta_{rs} \\ \delta_{lf} \\ \delta_{rf} \\ \delta_{rud} \end{bmatrix} + \begin{bmatrix} 14 & 0 & 0 & 0 & 0 \\ 0 & 14 & 0 & 0 & 0 \\ 0 & 0 & 14 & 0 & 0 \\ 0 & 0 & 0 & 14 & 0 \\ 0 & 0 & 0 & 0 & 14 \end{bmatrix} \begin{bmatrix} \delta_{ls-c} \\ \delta_{rs-c} \\ \delta_{lf-c} \\ \delta_{rf-c} \\ \delta_{rud-c} \end{bmatrix} \tag{3-11}$$

Equation (3-10) is then augmented to the state space model, (Equation (3-7)), yielding the augmented equation:

$$\begin{aligned} \dot{\mathbf{x}}_{aug}(t) &= \begin{bmatrix} \mathbf{A} & \mathbf{B}_{\mathrm{mod}} \\ 0 & -14\cdot\mathbf{I} \end{bmatrix} \mathbf{x}_{\mathbf{aug}}(t) + \begin{bmatrix} 0 \\ 14\cdot\mathbf{I} \end{bmatrix} \mathbf{u}_{\mathrm{aug}}(t) \\ &= \mathbf{A}_{aug}\cdot\mathbf{x}_{\mathbf{aug}}(t) + \mathbf{B}_{\mathbf{aug}}\mathbf{u}_{\mathrm{aug}}(t) \end{aligned} \tag{3-12}$$

with the augmented state and control input vectors: $\mathbf{x}_{aug}$, $\mathbf{u}_{aug}$, listed in Tables 3-5 and 3-6:

$\mathbf{x}_{aug}$	Augmented State Variables	Units
1	θ Pitch Angle	rad
2	u Forward velocity	ft/sec
3	α Angle of attack	rad
4	q Pitch rate	rad/sec
5	ϕ Bank angle	rad
6	β Sideslip angle	rad
7	p Roll rate	rad/sec
8	r Yaw rate	rad/sec
9	δ_{ls} Left stabilator position	rad
10	δ_{rs} Right stabilator position	rad
11	δ_{lf} Left flaperon position	rad
12	δ_{rf} Right flaperon position	rad
13	δ_{r} Rudder position	rad

Table 3-5 Augmented State Vector

u_{aug}	Augmented Control Inputs (Perturbations)	Units
1	δl_{ls-c} Left stabilator command	rad
2	δ_{rs-c} Right stabilator command	rad
3	δ_{lf-c} Left flaperon command	rad
4	δ_{rf-c} Right flaperon command	rad
5	δ_{r-c} Rudder command	rad

Table 3-6 Augmented Control Input Vector

3.7.3. *Discrete-time Measurement Models*

Since sensor measurements are sampled by the flight control system at *64 Hz* in the SRF simulation, the sensor measurement and output models are developed in discrete-time form. Assuming we wish to measure states and control surface positions, the measurement model is:

$$\mathbf{z}(t_i) = \begin{bmatrix}\mathbf{H} & \mathbf{D}_z\end{bmatrix}\mathbf{x}_{aug}(t_i) = \text{"}\mathbf{H}_{aug}\text{"}\mathbf{x}_{aug}(t_i) \tag{3-13}$$

As in prior research [14, 26, 45], the sensor measurements available are those needed as feedback signals by the Block 40 flight control system and which are listed in Table 3-7:

z	Measurement Variables	Units
1	α Angle of attack	rad
2	q Pitch rate	rad/sec
3	a_n Normal acceleration at pilot's station	g's
4	p Roll rate	rad/sec
5	r Yaw rate	rad/sec
6	a_y Lateral acceleration at pilot's station	g's

Table 3-7 Measurement Vector

3.7.4. *Acceleration Sensor Models.*

All output quantities and all but two measurement quantities are already available from the design model (Equation (3-13)). The two measurement quantities not available are normal and lateral acceleration at the pilot's station, a_n, and a_y. Eide [14] and Stepaniak [45] developed linear versions of the nonlinear a_n

and a_y sensor models used on the SRF. While the linear normal acceleration model gave a good representation of the perturbation normal acceleration, the linear model for lateral acceleration did not, even with measurement noise added [14]. Perhaps lacking a means to verify the a_y sensor truth model, Stepaniak elected to discard the nonlinear truth model a_y sensor, replace it with the linear version and continue research. This thesis will use those models, both of which are given in Equation (3-14):

$$\begin{aligned} a_n &= -\frac{u}{g}(\dot{\alpha} - q) + \frac{l_x}{g}\dot{q} \\ a_y &= -\frac{u}{g}(\dot{\beta} + r) - \phi + \frac{l_x}{g}\dot{r} \end{aligned} \tag{3-14}$$

where g = 32.17 *ft/sec*^{2} and l_x, the distance from the aircraft center of gravity to the pilot's station, is calculated using:

$$l_x = 9.988 + \bar{c} \cdot x_{cg} \cdot 0.01 \tag{3-15}$$

$\bar{c}$ is the mean aerodynamic chord (11.32 ft) and x_{cg} is, for this thesis, defined as the *average* center of gravity location along the x-body axis and equals 37.376 ft. Substituting these values into Equation (3-15) yields: l_x = 14.219 *ft*. By expressing $\dot{\alpha}$, $\dot{\beta}$, ṙ in terms of state variables (Eide [14]: pg. 51), the measurement equation ($\mathbf{z}(t_i) = \mathbf{H}\mathbf{x}(t_i) + \mathbf{D}_z\mathbf{u}_{\mathbf{mod}}(t_i)$, $= \begin{bmatrix}\mathbf{H} & \mathbf{D}_z\end{bmatrix}\mathbf{x}_{aug}(t_i)$, $= \mathbf{H}_{aug}\mathbf{x}_{aug}(t_i)$) is written as:

$$\begin{bmatrix} u \\ \alpha \\ q \\ a_n \\ p \\ r \\ a_y \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{\overline{U}}{32.2}[Z'_\theta & Z'_u & Z'_\alpha & Z'_q - 1 & 0 & 0 & 0 & 0] \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & [\frac{\overline{U}}{32.2}Y'_\phi - 1] & \frac{\overline{U}}{32.2}[Y'_\beta & Y'_p & Y'_r - 1] \end{bmatrix} \begin{bmatrix} \theta \\ u \\ \alpha \\ q \\ \phi \\ \beta \\ p \\ r \end{bmatrix} +$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -\frac{\overline{U}}{32.2}[.5Z'_{\delta_e} & .5Z'_{\delta_e} & .5Z'_{\delta_f} & .5Z'_{\delta_f} & 0] \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \frac{\overline{U}}{32.2}[-.5Y'_{\delta_{dt}} & .5Y'_{\delta_{dt}} & -.5Y'_{\delta_a} & .5Y'_{\delta_a} & Y'_{\delta_r t}] \end{bmatrix} \begin{bmatrix} \delta_{ls} \\ \delta_{rs} \\ \delta_{lf} \\ \delta_{rf} \\ \delta_{rud} \end{bmatrix} \tag{3-16}$$

where $\overline{U}$ is the forward velocity at trim, and all other quantities are defined in Table 3-2.

3.8. *Kalman Filter Failure Modeling: F_{ai} and F_{sj}*

It is appropriate at this point to explain and implement the method of modeling the hypothesized failures in the Kalman filters. Total actuator failures and partial control surface impairments are inserted into the *continuous-time* plant dynamics model (Equation (3-12)) via a "failure matrix", $\mathbf{F_{ai}}$, *before they are discretized*, corresponding to the insertion of impairments into the "continuous-time" truth model:

$$\dot{\mathbf{x}}_{aug}(t) = \mathbf{A}_{aug} \cdot \mathbf{x}_{\mathbf{aug}}(t) + \mathbf{B}_{\mathbf{aug}}\mathbf{F}_{\mathbf{ai}}\mathbf{u}_{\mathbf{aug}}(t) \tag{3-17}$$

where $\mathbf{F_{ai}}$ is an identity matrix under no-fail conditions, and $\mathbf{F_{ai}}$ (i, i) = ε; $0 \le \varepsilon \le 1$, if the i^{th} actuator is totally failed (ε=0) or partially impaired (0<ε<1). Sensor failures are modeled the same exact way, but using a sensor failure matrix, $\mathbf{F_{sj}}$, in the discrete-time measurement equations:

$$\mathbf{z}(t_i) = \mathbf{F_{sj}}\mathbf{H}_{aug}\mathbf{x}_{aug}(t_i) \tag{3-18}$$

where $\mathbf{F_{sj}}$ is *nominally* an identity matrix, but $\mathbf{F_{sj}}(j,j) = \varepsilon$; $0 \le \varepsilon \le 1$, if the j^{th} sensor is totally failed or partially impaired.

Dynamics and Measurement Noises

Now we will add noise to the continuous plant dynamics and discrete-time measurement equations to represent modeling uncertainties, turbulence, and sensor noise:

$$\dot{\mathbf{x}}_{aug}(t) = \mathbf{A}_{aug} \cdot \mathbf{x}_{aug}(t) + \mathbf{B}_{aug}\mathbf{F}_{ai}\mathbf{u}_{aug}(t) + \mathbf{G}_{aug}\mathbf{w}(t) \tag{3-19}$$

$$\mathbf{z}(t_i) = \mathbf{F}_{sj}\mathbf{H}_{aug}\mathbf{x}_{aug}(t_i) + \mathbf{v}(t_i) \tag{3-20}$$

$\mathbf{G}_{aug}$ is the augmented noise injection matrix:

$$\mathbf{G}_{aug} = \begin{bmatrix} \mathbf{G}_{8x6} \\ \mathbf{0}_{5x6} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ X'_u & X'_\alpha & X'_q & 0 & 0 & 0 \\ Z'_u & Z'_\alpha & Z'_q & 0 & 0 & 0 \\ M'_u & M'_\alpha & M'_q & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & Y'_p & Y'_\beta & Y'_r \\ 0 & 0 & 0 & L'_p & L'_\beta & L'_r \\ 0 & 0 & 0 & N'_p & N'_\beta & N'_r \\ & & & \left[\mathbf{0}_{5x6}\right] & & \end{bmatrix} \tag{3-21}$$

where $\mathbf{G}_{8x6}$ is the matrix necessary to inject noise into the continuous-time dynamic states: $u, \alpha, q, p, \beta, r$. The 5x6 null matrix, $\left[\mathbf{0}_{5x6}\right]$, shows that noise will not be injected into the actuator dynamic states. The dynamics driving noise vector, $\mathbf{w}(t)$, is assumed to be Gaussian and white over the frequency bandwidth of the (rigid) aircraft. Its strength, $\mathbf{Q}$, is listed in Table 3-8 [14, 26, 45]:

Q element	Parameter	Average Noise Strength
Q(1,1)	u	4.5 x 10^{-2} ft^2 rad/sec
Q(2,2)	α	3.0 x 10^{-6} rad^2 sec
Q(2,3)	α vs q	1.1 x 10^{-8} rad^2
Q(3,3)	q	1.5 x 10^{-6} rad^2/sec
Q(4,4)	p	6.0 x 10^{-6} rad^2/sec
Q(5,5)	β	3.0 x 10^{-6} rad^2 sec
Q(5,6)	β vs r	6.3 x 10^{-9} rad^2
Q(6,6)	r	2.4 x 10^{-6} rad^2/sec
Note: All elements not specifically listed are assumed to be zero.		

Table 3-8 Dynamics Driving Noise Strength

Discrete-time measurement noise $\mathbf{v}(t_i)$ is also assumed to be white and Gaussian, due to the very large sensor noise bandwidth when compared to the bandwidth of the rigid aircraft. Table 3-9 lists the strength, $\mathbf{R}$ [45]:

R element	Parameter	RMS Noise Strength
R(1,1)	α	0.004 *rad*
R(2,2)	q	0.006 *rad/sec*
R(3,3)	a_n	0.01 *g's*
R(4,4)	p	0.02 *rad/sec*
R(5,5)	r	0.006 *rad/sec*
R(6,6)	a_y	0.005 *g's*

Table 3-9 Sensor Measurement Noise Standard Deviations

We have now completely defined the following set of stochastic, continuous-time plant and discrete-time measurement state equations:

$$\begin{aligned}\dot{\mathbf{x}}_{aug}(t) &= \begin{bmatrix}\mathbf{A} & \mathbf{B}_{\text{mod}} \\ \mathbf{0} & -14\cdot\mathbf{I}\end{bmatrix}\mathbf{x}_{\mathbf{aug}}(t)+\begin{bmatrix}\mathbf{0} \\ 14\cdot\mathbf{I}\end{bmatrix}\mathbf{F}_{\mathbf{ai}}\mathbf{u}_{\text{aug}}(t)+\begin{bmatrix}\mathbf{G} \\ \mathbf{0}\end{bmatrix}\mathbf{w}(t) \\ &= \mathbf{A}_{aug}\cdot\mathbf{x}_{\mathbf{aug}}(t)+\mathbf{B}_{\mathbf{aug}}\mathbf{F}_{\mathbf{ai}}\mathbf{u}_{\text{aug}}(t)+\mathbf{G}_{aug}\mathbf{w}(t)\end{aligned} \tag{3-22}$$

$$\mathbf{z}(t_i) = \mathbf{F}_{\mathbf{sj}}\mathbf{H}_{aug}\mathbf{x}_{aug}(t_i)+\mathbf{v}(t_i) \tag{3-23}$$

The last step in deriving the MMAE Kalman filter design model is to discretize the continuous-time dynamics model to reflect the fact that the MMAE algorithm would be implemented on a digital flight

control computer. In this thesis, the discretization is performed assuming a fixed sample rate of 64 Hz, so the resulting state transition and discrete control matrices, Φ and $\mathbf{B}_d$, are [14]:

$$\Phi = e^{\mathbf{A}_{aug}\Delta T}$$

$$\mathbf{B}_d = \left(\int_0^{\Delta T} e^{\mathbf{A}_{aug}\tau} d\tau \right) \mathbf{B}_{aug} \mathbf{F}_{ai} \tag{3-24}$$

where ΔT is the sampling period of $1/64$ th-second. The appropriate discrete-time white noise, $\mathbf{w}_d$, has the following statistics [14]:

$$\begin{aligned} &E\{\mathbf{w}_d(t_i)\} = \mathbf{0} \\ &E\{\mathbf{w}_d(t_i)\mathbf{w}^T{}_d(t_i)\} = \mathbf{Q}_d = \int_0^{\Delta T} e^{\mathbf{A}_{aug}\tau} \mathbf{G}_{aug} \mathbf{Q} \mathbf{G}_{aug}{}^T e^{\mathbf{A}^T{}_{aug}\tau} d\tau \\ &E\{\mathbf{w}_d(t_i)\mathbf{w}^T{}_d(t_j)\} = \mathbf{0}, \quad t_i \neq t_j \end{aligned} \tag{3-25}$$

The resulting equivalent discrete-time system description is thus:

$$\mathbf{x}_{aug}(t_{i+1}) = \Phi\, \mathbf{x}_{aug}(t_i) + \mathbf{B}_d \mathbf{u}_{aug}(t_i) + \mathbf{w}_d(t) \tag{3-26}$$

$$\mathbf{z}(t_i) = \mathbf{F}_{sj} \mathbf{H}_{aug} \mathbf{x}_{aug}(t_i) + \mathbf{v}(t_i) \tag{3-27}$$

This completes MMAE model development. The next section discusses the concept of *Control Redistribution* in detail, starting with some interesting history and then reviewing the Control Redistribution algorithm as developed independently by Stepaniak [45].

3.9. *Control Redistribution*

This section reviews Stepaniak's [45] independent 1995 development of Control Redistribution (CR). Stepaniak developed CR when it became apparent that MMAC would not perform as expected, due

to numerical problems which occurred during LQ controller synthesis. Control Redistribution is a simple, intuitive concept that is easily explained using linear algebra equations, and has the added benefit that it is a passive, drop-in module which does not alter the existing Block 40 flight control system in any way. It merely operates on, or *redistributes* Block 40 commands in an attempt to restore control power lost when one or more control surfaces are impaired. It turns out that this concept has a rich history behind it, but under other names, and this section will supply some of that missing history [7, 8, 11, 12, 15]. Section 3.9.2 will then repeat Stepaniak's excellent development and explanation of CR, followed in Section 3.9.3 by an examination of the signal interface between CR and MMAE necessary for this thesis study of partial impairments.

3.9.1. Control Redistribution: History

The Air Force's "Reliability & Maintainability 2000" ("R&M 2000") program of the 1980's had many "thrusts", one of which was the development of "Fault-Tolerant" (Reliable) and less complex (maintainable) flight control systems, since it had been calculated [8, 15] that about 20% of aircraft losses in the Vietnam conflict were due to flight control system malfunctions. This need was the driving force behind the study of "Control Surface Reconfiguration", begun in Wright Laboratory in the early 1980's. "In-house" research into Control Surface Reconfiguration followed two paths, simulation and actual flight-testing. The simulation path consisted of non-real-time, "proof-of-concept" simulations done on the SRF, and real-time, man-in-loop simulations of promising concepts. Two real-time, motion simulations of note, performed in Wright Laboratory in 1987 and 1989 respectively, were the "AFTI F-16 Control Mixer Reconfiguration" and the "Control Reconfigurable Combat Aircraft" (CRCA) simulations [11, 12].

The AFTI simulation tested a detection and reconfiguration concept developed by General Electric [18]. A module called "System Impairment Detection and Classification" (SIDC) used a battery of Log Likelihood and Sequential Probability Ratio Tests to perform detection of "stuck, floating, or partially missing" control surfaces. The "Control Mixer" was GE's name for their reconfiguration module. It took the identification of the impaired control surface, along with the degree of surface impairment (amount lost) from SIDC, and reconfigured the remaining surfaces to restore lost control power. When reviewed in the

proper context, results were dramatic! SIDC failed miserably at first, probably because it required tuning and insight (beyond the capability of in-house engineers [12]) to eliminate missed and false alarms. The reconfiguration module, however, when supplied with perfect, instant impairment information, performed beautifully. It did indeed restore lost stability and control power, enabling pilots to track a maneuvering target under surface impairments causing PIO and aircraft departure when reconfiguration was removed. Since this simulation incorporated a feature which computed linearized stability and control derivatives continuously in real-time, the reconfiguration module operated throughout the AFTI F-16 flight envelope [11]. As a result, reconfiguration was studied even in landing conditions, where it was seen (as predicted later by Stepaniak [45]) that during more severe impairments of multiple surfaces, reconfiguration might indeed drive remaining surfaces into rate or position saturation, causing departure. SIDC performance was eventually improved to about a 50% success rate by removing the linear aircraft models and stochastic noises, and putting the actual truth model equations of motion in their place. While perhaps not a valid detection scheme, SIDC in the loop with reconfiguration did allow the (full envelope) study of minimum detection times and transients occurring during reconfiguration (which were negligible). It was also seen that, during pilot maneuvering or during turbulence, pilot induced dithering of the control surfaces was unnecessary for impairment detection. This scheme by General Electric was actually flight tested aboard the HIDEC F-15 [7, 38], which, since it possessed a digital, fly-by-wire flight control system, was capable of hosting the algorithm. The control surface "failure" was a complete loss of one stabilator, simulated by commanding it to equal the local angle of attack. Test pilots flying the aircraft could easily distinguish the difference between the non-reconfigured and reconfigured aircraft, since aircraft handling qualities improved remarkably when control reconfiguration, or redistribution, was added [7, 38].

The Control Reconfigurable Combat Aircraft (CRCA) Simulation tested a competing detection and reconfiguration algorithm developed by Grumman Aircraft Corporation, Lear Astronics, and Charles River Analytics [6, 21, 22]. The primary goal of this project was to create a "Control Reconfiguration Handbook" for designers of future aircraft to follow in "designing in" control redundancy, and also to explore the cockpit displays a pilot might need to be warned of impairments and of resulting new limitations on aircraft performance. Results of this ambitious project were less than originally hoped. The problem was that, while the host aircraft was an advanced combat aircraft design from NASA fully supported by wind tunnel data, it

lacked a full-envelope flight control system with "Level 1" performance against which to measure baseline flying and handling qualities. The flight control system, and hence the simulation, was valid at only a handful of discrete flight conditions, and pilot test subjects could neither hold those conditions accurately nor could they discern any positive effects due to reconfiguration [12].

Other laboratory projects included flight tests of control surface reconfiguration aboard Flight Dynamics Laboratory's Unmanned Research Vehicle [12]. In one particular test, an aileron was physically ejected from the aircraft and the effects of reconfiguration were evaluated by the URV "pilot" and from telemetry data.

Though none of these laboratory projects reached their full data-yielding potential, two points were clear from the data which was acquired. The first was that, *assuming an accurate detection*, control surface reconfiguration was a demonstrable benefit, with the potential to save lives and aircraft [11]. The second point was that an accurate, reliable fault detection and isolation algorithm had yet to be found. Perhaps MMAE is that algorithm.

3.9.2. Control Redistribution Algorithm Development

Control Redistribution uses *inherent redundancy* existing in an aircraft's suite of control effectors so that, if an impairment to a control surface does occur, commands to that surface can be "re-routed" to other surfaces and any control power lost can be restored. Mathematically, this says that aircraft performance before and after a control surface impairment should be equal:

$$\begin{aligned} \dot{\mathbf{x}}_{\textit{before}\ \text{impairment}} &= \dot{\mathbf{x}}_{\textit{after}\ \text{impairment}} \\ (\mathbf{Ax}+\mathbf{Bu}) &= (\mathbf{A}_{\mathbf{fail}}\mathbf{x}+\mathbf{B}_{\mathbf{fail}}\mathbf{u}_{\mathbf{r}}) \end{aligned} \tag{3-28}$$

It is also assumed that when a control surface impairment occurs, only the aircraft control matrix, **B**, is affected, and any changes to the stability matrix **A** are assumed to be negligible:

$$\mathbf{A} \approx \mathbf{A}_{\textit{fail}} \tag{3-29}$$

hence, the Control Redistribution technique is based on equating the redistributed control input of a failed system to that of a fully functional system:

$$\mathbf{B}_{fail}\mathbf{u}_r \approx \mathbf{Bu} \tag{3-30}$$

where $\mathbf{B}_{fail}$ represents the input matrix corresponding to a failed actuator condition and $\mathbf{u}_r$ represents the redistributed control signals. The failed input matrix, $\mathbf{B}_{fail}$ is expressed as:

$$\mathbf{B}_{fail} = \mathbf{BF}_{ai} \tag{3-31}$$

where $\mathbf{F}_{ai}$ is the same actuator failure matrix used in Kalman filter development (Section 3.8). The vector of redistributed control inputs, $\mathbf{u}_r$, is obtained by multiplying the normal (unimpaired) control vector, $\mathbf{u}$, by a *control redistribution matrix* for the i^{th} actuator, $\mathbf{D}_{ai}$:

$$\mathbf{u}_r = \mathbf{D}_{ai}\mathbf{u} \tag{3-32}$$

$\mathbf{D}_{ai}$ must be computed either on-line or a-priori (if possible) for each of the "i"actuators as follows:

Substituting Equations (3-31) and (3-32) into Equation (3-30) yields:

$$\mathbf{BF}_{ai}\mathbf{D}_{ai}\mathbf{u} = \mathbf{Bu} \tag{3-33}$$

which is true for any general control input, $\mathbf{u}$. Removing $\mathbf{u}$ from both sides of Equation (3-33) yields:

$$\mathbf{BF}_{ai}\mathbf{D}_{ai} = \mathbf{B} \tag{3-34}$$

Since $\mathbf{F}_{ai}$ *may* have one or more columns of zeroes (corresponding to one or more completely failed actuators), we cannot remove $\mathbf{B}$ from each side of Equation (3-34) because $\mathbf{B}$ and $\mathbf{F}_{ai}$ become, in effect, "welded" together to form $\mathbf{B}_{fail}$. If this is the case, the matrix $\mathbf{BF}_{ai}$ is rank deficient and the solution for the redistribution matrix $\mathbf{D}_{ai}$ becomes:

$$\mathbf{D}_{ai} = (\mathbf{BF}_{ai})^{+}\mathbf{B} \tag{3-35}$$

The superscript '+' indicates the Moore-Penrose pseudoinverse [47]. The pseudoinverse has two desirable mathematical properties. The first is that it yields a true inverse where possible, for example, in cases of partial impairments when $\mathbf{F}_{ai}$ maintains full-rank:

$$\mathbf{D}_{ai} = (\mathbf{BF}_{ai})^{+}\mathbf{B}, \; = (\mathbf{BF}_{ai})^{-1}\mathbf{B}, \; = (\mathbf{F}_{ai}{}^{-1})(\mathbf{B})^{-1}\mathbf{B}, \; = \mathbf{F}_{ai}{}^{-1} \tag{3-36}$$

The second property is that in cases (such as total impairments) when a true inverse is not possible, the pseudoinverse yields the best solution in the least squares sense, i.e. the error, $\|\mathbf{BF}_{ai}\mathbf{D}_{ai} - \mathbf{B}\|$ is minimized [47]. In other words, Equation (3-30) is satisfied *as closely as possible.*

It is instructive to note the form of the control redistribution matrix, $\mathbf{D}_{ai}$. Under nominal conditions with no impairments, it is an identity matrix which passes surface commands through unaltered. When the i^{th} actuator is totally failed, $\mathbf{D}_{ai}$ has the same form of an identity matrix, except that the i^{th} column is replaced with non-zero entries which effectively redistribute commands meant for the i^{th} actuator to the other actuators. An example of a $\mathbf{D}_{ai}$ matrix corresponding to a totally failed right stabilator (i=2) is given in Equation (3-37), along with the full, "prepacked" $\mathbf{D}_{ai}$ matrix. It is apparent that $\mathbf{D}_{a2}$ is obtained by replacing the 2nd column of a 5x5 identity matrix with the 2nd column of $\mathbf{D}_{ai}$:

$$\mathbf{D}_{ai} = \begin{bmatrix} \mathbf{D}_{a1} & \mathbf{D}_{a2} & \mathbf{D}_{a3} & \mathbf{D}_{a4} & \mathbf{D}_{a5} \\ 0 & 1 & 1.1037 & -1.1037 & -1.2719 \\ 1 & 0 & -1.1037 & 1.1037 & 1.2719 \\ 0.9060 & -0.9060 & 0 & 1 & 1.1524 \\ -0.9060 & 0.9060 & 1 & 0 & -1.1524 \\ -0.7862 & 0.7862 & 0.8678 & -0.8678 & 0 \end{bmatrix} \qquad \mathbf{D}_{a2} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -0.9060 & 1 & 0 & 0 \\ 0 & 0.9060 & 0 & 1 & 0 \\ 0 & 0.7862 & 0 & 0 & 1 \end{bmatrix} \tag{3-37}$$

The matrix $\mathbf{D}_{a2}$ shows that all actuators *other* than the right stabilator (i≠2) receive their normal commands, represented by the diagonal ones in the matrix. However, note from column 2 that the command to the totally-failed right stabilator, $\mathbf{D}_{a2}(2,2)$, is zero, and that the other actuators are now receiving additional commands to try to restore control authority lost from the failure. Specifically, the left stabilator now receives the entire right stabilator command (*in addition to its nominal command*), and the flaperons and rudder act as necessary to restore the rolling moment normally produced by the (now failed) right stabilator actuator.

In this thesis research exploring conditions of *partial* impairments to the i^{th} actuator, $\mathbf{F}_{ai}(i,i) = \varepsilon \; ; \; 0 < \varepsilon \leq 1$, and $\mathbf{D}_{ai}$ is an identity matrix except that $\mathbf{D}_{ai}(i,i) = 1/\varepsilon$. To alter the example given above, if the right stabilator is instead 50% impaired, $\varepsilon = .5$, $\mathbf{D}_{ai}$ just equals $\mathbf{F}_{ai}^{-1}$ and:

$$\mathbf{D}_{ai} = \mathbf{D}_{a2} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3-38}$$

which makes intuitive sense, for if the stabilator is 50% missing, the algorithm commands the surface to twice the deflection. Although this may appear to be the best solution, additional thought reveals that it is not necessarily, for if the right stabilator were 80% missing, ε would equal .2 and $\mathbf{D}_{ai}(2,2)$ would equal 5. Commanding a surface by a factor of five times the normal is unacceptable, given the actuator rate and position limits which would undoubtedly occur, especially during aircraft maneuvering. Maybeck [37] has suggested an alternative to computing and using $\mathbf{F}_{ai}^{-1}$, in cases of partial impairments, which would simplify redistribution calculations and perhaps prevent rate and position saturations of remaining actuators. Since the test cases are performed without maneuvers in this thesis, we *will* use "true inverse" calculations, noting that there *is* an alternative if it is needed (see: Appendix M for a description of the method).

3.9.3. Interface with MMAE

Previous research with Control Redistribution [26, 45] investigated *single* and *dual, total* actuator failures in which it was possible to "pre-define", or "prepack" Control Redistribution (CR) matrices, $\mathbf{D}_{ai}$, (Equation (3.37)) and insert them once MMAE made the identification, ('i') of the impaired actuator. This was a "soft interface" between MMAE and the Control Redistribution Module, in that probabilities from MMAE did *not* enter into CR calculations *directly*, but rather, by surpassing a set threshold value, served as a "trigger" to start control reconfiguration based on prior assumptions of *totally failed* actuators. For this

thesis project, however, a "hard interface" between MMAE and CR is necessary. For cases of *partial* actuator impairment, estimates from each of 12 "channels" of the MMAE algorithm in the form of conditional probability values, $p_k(t_i)$, *directly influence* the calculation of the Control Redistribution matrix, $\mathbf{D_{ai}}$:

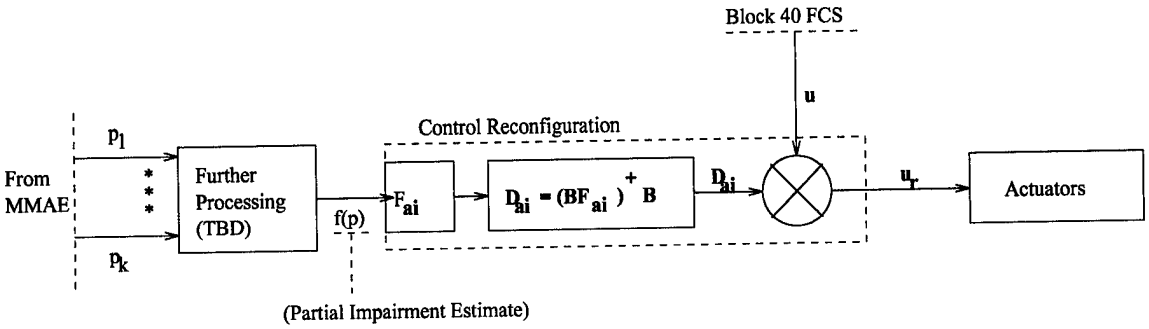


Figure 3-2 MMAE Interface With Control Redistribution

In Figure 3-2, "$\mathbf{F_{ai}}$" is the failure matrix into which either the "raw conditional probabilities", or values which are *functions* of MMAE conditional probabilities (to obtain more accurate estimates of partial actuator impairment), are inserted. Equation (3-39) shows how these values enter into Control Reconfiguration calculations. While in theory, *all* diagonal entries *could* be values other than 1, we will only have a maximum of two actuator impairments at a time in this thesis research, and hence, only two diagonal entries which are not equal to 1:

$$\mathbf{F_{ai}} = \begin{bmatrix} 1-f(p_{k_{sl}}) & 0 & 0 & 0 & 0 \\ 0 & 1-f(p_{k_{sr}}) & 0 & 0 & 0 \\ 0 & 0 & 1-f(p_{k_{fl}}) & 0 & 0 \\ 0 & 0 & 0 & 1-f(p_{k_{fr}}) & 0 \\ 0 & 0 & 0 & 0 & 1-f(p_{k_{rud}}) \end{bmatrix} \tag{3-39}$$

It is important to note from Figure 3-2 that the "hard interface" between MMAE and Control Reconfiguration may consist of measures (described in Chapter 4) to process the raw conditional probabilities from MMAE into a more usable form by the implementation of a "smoothing and quantization module". Such a module would convert constantly, time-varying probability values from MMAE into more

steady, piecewise-linear, or "quantized" estimates, thus precluding the undesirable, continuous recomputation of $\mathbf{D}_{ai}$ and continuous reconfiguration. The requirement for and design of such an "interface module" will be explained fully in Chapter 4.

3.10. Chapter Summary

This chapter began with a short history of the Simulation Rapid Prototyping Facility, or SRF. An introduction to the VISTA F-16 aircraft was given next and followed by a description of the non-realtime simulation of that aircraft on the SRF. The method of simulating total actuator failures and partial losses of control surface effectiveness, or "soft failures", within the SRF/VISTA F-16 truth model was then discussed. The chapter then explained the construction of linear, reduced order aircraft, actuator, and measurement models used by the Kalman Filters in the MMAE algorithm and the failure modeling method used. Control Redistribution (CR) was explained, including history, algorithm development, and its interface with MMAE and the VISTA flight control system.

4. *Simulation Results*

4.1. Chapter Overview

This chapter begins by examining MMAE performance during partial impairments of single actuators. First, the definition and method of inserting partial impairments into the VISTA F-16 truth model is presented and Stepaniak's [45] results are duplicated in order to validate the new method of modeling partial impairments. An extensive battery of single-surface, partial actuator impairments is then performed and MMAE probability plots are examined to gain insight both into MMAE detection and blending capability and into solutions to anticipated interface problems between MMAE and Control Reconfiguration (CR).

Many issues are uncovered as a result of these single surface tests, such as the need for "probability smoothing" and "probability quantization" to obtain a suitable interface between MMAE and Control Reconfiguration (to prevent the reconfiguration declaration from changing too quickly or erratically), and also to reduce the number of additional, "partial fail filters" to be designed and implemented within the MMAE algorithm. The need to retune the MMAE for partial rudder impairments is made clear, and some possible solutions to this issue are presented.

After the MMAE algorithm is expanded to include the new banks of partial-fail filters and the interface between MMAE and Control Reconfiguration (CR) is defined and implemented, the first tests of dual, total actuator impairments and actuator-sensor combinations with reconfiguration are completed and the results are presented and critiqued. These results are followed by tests of dual actuator impairments of 75% (25% effectiveness), and combinations of 75% actuator impairments and 100% sensor impairments with Control Reconfiguration. Results of these tests are presented and critiqued, and resulting MMAE/CR performance issues are highlighted as well as suggestions made to solve them. Possible problems with Control Reconfiguration performance are found and presented as an important course of future research. The last group of partial impairment tests studied in this thesis are dual actuator impairments of 50% (50%

effectiveness) and "50% actuator, 100% sensor impairment" combinations with Control Reconfiguration. These results, though disappointing, are presented to understand the degradation in MMAE performance and the means to improve performance. The chapter ends with a summary.

4.2. *MMAE Detection Performance during Partial Impairments of Single Actuators*

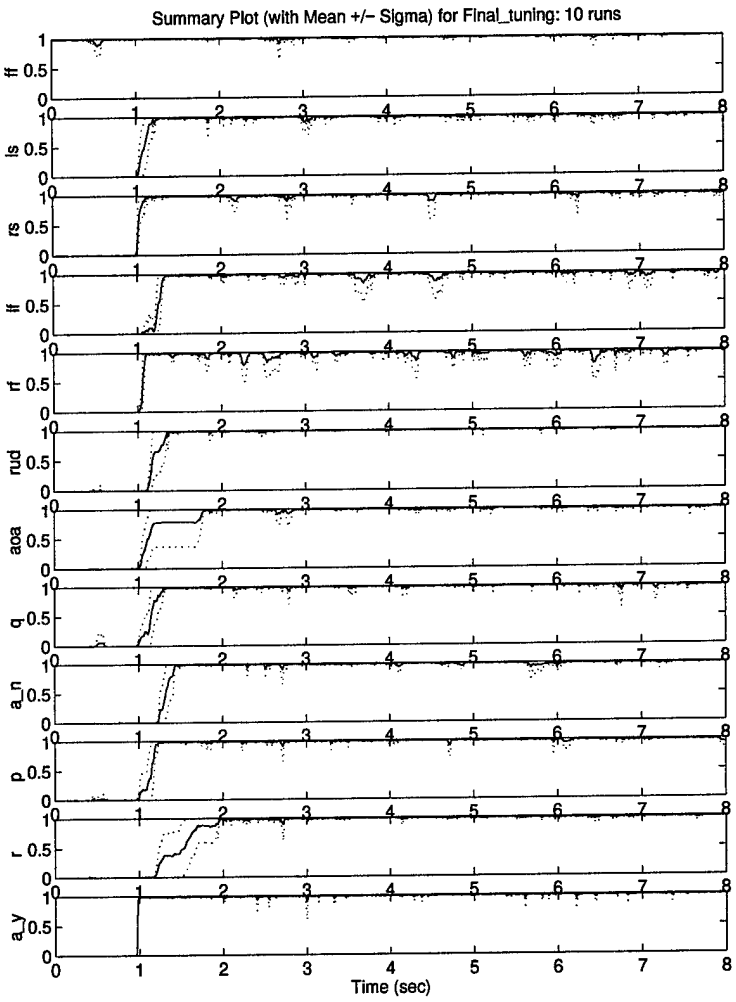


Figure 4-1 MMAE Single Impairment Detection Performance:

No Maneuver, No Reconfiguration, Stepaniak Dither

This phase of thesis research was necessary : 1) to define, implement, and test the method of inserting partial actuator impairments, 2) to gain insight into MMAE detection and blending capability

during partial actuator impairments, 3) to define the additional banks of "partial-failure filters" needed for the study of dual, partial actuator impairments to come later and, 4) to devise an appropriate interface between MMAE and Control Reconfiguration, since the method used in Stepaniak [45] and Lewis [26] for total actuator failures is not adequate for partial actuator impairments.

The starting point for this thesis research was chosen to be the reproduction of Stepaniak's [45] thesis results for total impairments of single actuators and sensors, for cases of "no-maneuvers" and without Control Reconfiguration, and using Stepaniak's refined control dither scheme [45]. His results were quickly duplicated and are shown in Figure 4-1.

4.3. *Method of Partial Impairment Simulation and Insertion*

Both Stepaniak [45] and Lewis [26] simulated total actuator failures in the VISTA F-16 truth model by overwriting the actuator position, of the affected actuator, with the trim position at the time of failure. Before time of actuator failure:

$$\delta_{act} = \frac{(20.2)(144.8)(71.4)^2 \delta_{com}}{(s+20.2)(s+144.8)(s^2+2(0.736)(71.4)s+71.4^2)} \tag{4-1}$$

At the instant of actuator failure:

$$\delta_{act} = \delta_{trim} \tag{4-2}$$

Though such an instantaneous jump in actuator position is not realistic, it is not totally unreasonable since surface deflections are only a few degrees. This thesis models partial actuator impairments in such a manner that, for a continuous range of effectiveness factor, ε; $0 \le \varepsilon \le 1$, the impaired actuator position takes on values from "stuck" at the trim position, corresponding to $\varepsilon = 0$ and a total actuator impairment, to the fully operational actuator deflection when $\varepsilon = 1$:

$$\delta_{act} = \varepsilon\delta_{act} + (1-\varepsilon)\delta_{trim} \tag{4-3}$$

Again, though not a realistic way to model an actuator impairment, the above method is "what the MMAE detection algorithm is tuned to see" (at this time), and is not totally unreasonable. This method of modeling partial failures was validated, for the purpose of this thesis research, by setting $\varepsilon = 1$ and $\varepsilon = 0$ and rerunning Stepaniak's fully functional and total failure cases and noting perfect agreement between results.

4.4. MMAE Detection and Blending Capability during Single Partial Actuator Impairments

The next task was to examine the ability of the MMAE algorithm, well-tuned by Stepaniak [45] for detecting *total* actuator failures, to blend the correct, fully-failed actuator channel with the fully-functional aircraft channel in the MMAE filter bank to obtain an accurate estimate of a given *partial* actuator impairment. An idealized picture showing *desired* MMAE performance is given in Figure 4-2:

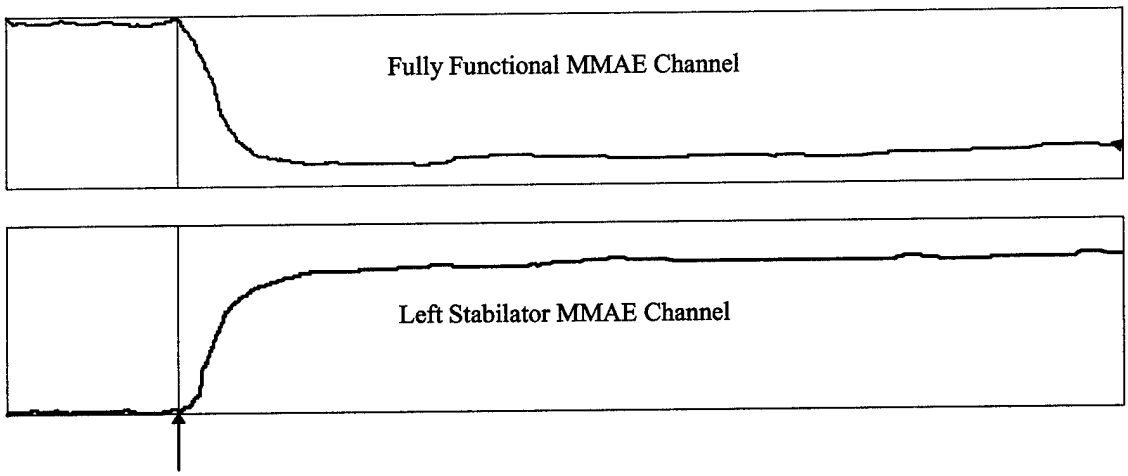


Figure 4-2 Idealized MMAE Blending Performance for 75% Left Stabilator Impairment

Note that at the time of a 75% impairment to the left stabilator at 1 second, the fully functional and the fully-failed left stabilator probabilities make smooth, rapid transitions down to .25 and up to .75, respectively. The probabilities remain locked at their new values and all other probabilities remain at zero value for the duration of the simulation. Furthermore, the probability of left stabilator failure equal to .75

provides, in this perfect example, a value of left stabilator effectiveness (1- .75 = .25) to supply directly to the Control Reconfiguration scheme to use in its computations.

4.5. Single Actuator Partial Impairment Test Cases

The list of single actuator, partial impairment test cases for the initial MMAE blending performance study is given in Table 4-1. The cases for $\varepsilon = 1$ (fully functional) and $\varepsilon = 0$ (fully failed) are not shown in the table but were performed as described in Section 4.2 and were identical to Stepaniak's results. To compare with previous research efforts, all impairments were inserted into the VISTA F-16 truth model at 1 second, and each simulation run was 8 seconds long. Each test point was repeated 10 times with different noise samples to produce a ten-run Monte Carlo analysis, as in all previous research, for statistical significance. The impairment tests were performed without Control Reconfiguration (CR) engaged and without any aircraft maneuvering, in order to establish a baseline judgment of MMAE blending performance against which to compare performance *incrementally* as CR and maneuvers are introduced (the latter to be accomplished by future research). It was desired to explore the high and low resolution blending ability of MMAE during these initial studies, hence, the values of Actuator Effectiveness Factor, ε, were chosen with smaller increments for the range: $0.0 \le \varepsilon \le 0.25$, and larger increments of 0.25 in the range: $0.25 \le \varepsilon \le 1.0$.

MMAE PERFORMANCE STUDY FOR SINGLE CONTROL SURFACE PARTIAL IMPAIRMENTS: NO RECONFIGURATION, NO MANEUVER, DITHER 'ON'						
Control Surface	Actuator Effectiveness Factor, ε, at 1 second (0 = totally failed, 1= fully functional)					
Left Stabilator	.05	.1	.15	.25	.50	.75
Right Stabilator	.05	.1	.15	.25	.50	.75
Left Flaperon	.05	.1	.15	.25	.50	.75
Right Flaperon	.05	.1	.15	.25	.50	.75
Rudder	.05	.1	.15	.25	.50	.75

Table 4-1 Single Control Surface Impairment Cases: No Reconfiguration, No Maneuver, Dither 'ON'

4.5.1. *Results of Section 4.5 and Discussion*

Probability Summary Plots of the results corresponding to the cases of Table 4-1 are found before the Appendices in Section SUM. Each Probability Summary Plot corresponds to a column of Table 4-1 and shows 6 subplots. The first subplot is the result of 10 Monte Carlo (MC) simulation runs on the "fully functional" (ff) channel *when no failures are inserted* , the second subplot is the result of 10 MC runs on the left stabilator (ls) channel when left stabilator impairments are inserted, the third subplot is the result of 10 MC runs on the right stabilator (rs) channel when right stabilator impairments are inserted, the fourth and fifth subplots are the result of 10 MC runs on the left flaperon (lf) and right flaperon (rf) channels when left and right flaperon impairments are inserted, and finally, the sixth subplot is the result of 10 MC runs on the rudder (rud) channel when rudder impairments are inserted. To get an accurate picture of probability "spillage", or activity onto other actuator and sensor channels during a given impairment condition, one must examine the Probability Plot (not the Probability Summary Plot) for that particular surface impairment. An example of a Probability Plot for the 75% left stabilator, partial impairment case is given in Figure 4-3, and Table 4-2 lists all MMAE channel abbreviations and their meanings. The Probability Plot shows all 12 MMAE channels. We see probability activity occurring primarily on the fully functional (ff) and left stabilator (ls) channels, as it should be, but we also see minor but noticeable probability "spillage" onto the right stabilator (rs), the rudder actuator (rud), angle-of-attack sensor (aoa), pitch rate sensor (q), and roll (p) and yaw rate (r) sensor channels:

MMAE Channel Abbreviation	Filter Hypothesis (Actuators)	MMAE Channel Abbreviation	Filter Hypothesis (Sensors)
FF	Fully Functional	AOA	Angle-of-Attack
LS	Left Stabilator	Q	Pitch Rate
RS	Right Stabilator	A_n	Normal Acceleration
LF	Left Flaperon	P	Roll Rate
RF	Right Flaperon	R	Yaw Rate
RUD	Rudder	A_y	Lateral Acceleration

Table 4-2 MMAE Filter Channel Abbreviations

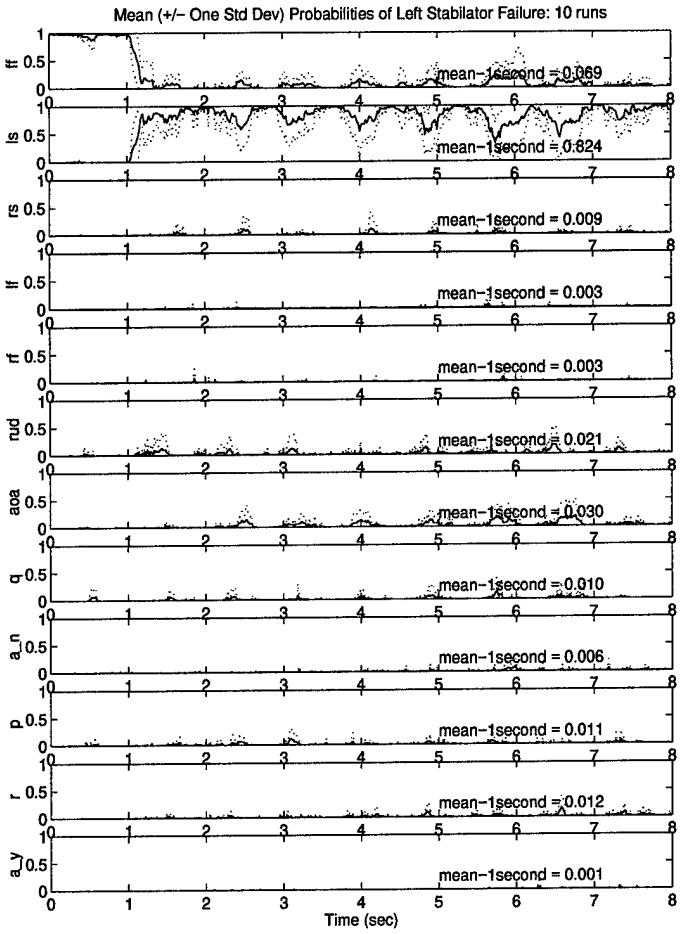


Figure 4-3 Probability Plot for 75% Left Stabilator Impairment

Upon examining the Probability Summary Plot in Figure 4-2 or *any* Probability Plot in Appendices A.1 through D.3, it is apparent immediately that the "ideal MMAE performance" as outlined earlier is not going to be attained. The probability values seen on the summary plots do not display the smoothness of Figure 4-2 or any stable, steady-state value nearly equal to the actual degree of actuator impairment, as we had hoped. Except for the rudder, which will be addressed separately, the Probability Summary Plots have "jagged edges" in general and, for ε = [.05,...,.25], bounce with strong periodic behavior between values greater and less than the value of the actual partial impairment. For ε = .5, the probability values (except for the rudder) oscillate between upper and lower values, both of which are lower than the actual value of impairment. Specifically, the left and right stabilators oscillate between upper and lower probability values of .45 and 0, while values for left and right flaperons are small enough most of the time that fully functional behavior is approximated. For ε = .75, MMAE does not detect *any* partial actuator impairments other than

a slight indication of partial impairment of the rudder. For reasons yet unknown, the rudder is strongly detected by MMAE as "fully failed" for ε between 0 and .5, but between ε =.5 and ε =.75, the algorithm apparently starts to blend estimates, rapidly approaching an indication of "fully functional". This is troublesome behavior which demanded special attention to be described shortly. Not seen in the summary plots, but which can be examined in the probability plot for each surface impairment condition, is the amount of blending between the fully functional aircraft (ff) MMAE probability plot, or "channel", and the pertinent channel for the partially impaired actuator. Ideally (and ignoring the negligibly small effect of the lower bounds on probabilities), the "fully functional" (ff) channel should decrease to a steady state value equal to :

$$p_{\textit{fully functional channel}} = 1 - p_{\textit{impaired actuator channel}} \tag{4-4}$$

with no probability "spillage" or "bleed-over" onto any other channels. Upon review of all probability plots for all test cases outlined in Table 4-1, it is clear that MMAE blending performance between "fully functional aircraft" (ff) and appropriate impaired actuator channels does not occur in the proportions desired. For surfaces other than the rudder, there is some activity on the "fully functional" probability plot, but there is also equivalent or greater probability activity spillage onto channels belonging to unimpaired actuators and sensors. The rudder is solidly declared by MMAE to be a total impairment for cases where the actual impairment ranges from 100% (total) to 50% (ε = [0,.05,...,.5]), with no probability blending or spillage whatsoever.

Despite the actual characteristics seen in the Probability Summary Plots, there are useful trends observed for all actuators *except* the rudder. A "time-average of the probability plot mean values" (a horizontal line approximation) can be fitted to the probability plots, starting at the time of impairment (1 second) and terminating at the simulation endtime (8 seconds), which approximates the "real" degree of partial impairment. Mean values of the probability plots for cases of ε = .1, .15, .25, and .5, for each actuator starting at 1 second and ending at 8 seconds, are summarized below and compared with the corresponding total failure case ($\varepsilon = 0$). Listed in the bottom row is the actual, or "Ideal Mean" value we would wish to see for the given impairment.

	Total Failure	90% Failure	85% Failure	75% Failure	50% Failure
Actuator	$\varepsilon = 0$	$\varepsilon = 0.10$	$\varepsilon = 0.15$	$\varepsilon = 0.25$	$\varepsilon = 0.50$
Left Stabilator	0.962	0.945	0.925	0.824	0.141
Right Stabilator	0.970	0.956	0.941	0.874	0.122
Left Flaperon	0.936	0.915	0.898	0.810	0.036
Right Flaperon	0.947	0.928	0.906	0.774	0.018
Rudder	0.960	0.957	0.954	0.948	0.900
"Ideal Value" $(1-\varepsilon)$	1.000	0.900	0.850	0.75	0.500

Table 4-3 Time-Averaged Probability Values for [1-8] seconds

Table 4-3 shows a proper downward trend in mean probability values with lesser degrees of impairment, or increasing values of actuator effectiveness, ε. Notice that the mean values for total failures, $\varepsilon = 0$, tend to be lower than the ideal value of 1.0. Aside from imperfect MMAE tuning, recall that another reason for this is that all probabilities are lower-bounded to prevent zero values and resulting lock-out. They are then rescaled to ensure a total summation to one, which means that no probability will ever achieve a value equal to one. In fact, with 12 elemental filters in the MMAE, each with a lower bound of 0.001, the closest any one probability could ever get to the ideal value of 1.000 would be 0.989 (i.e., the probabilities for all other 11 filters being at their lower bounds). Notice next that for values of effectiveness, ε, equal to 0.10, 0.15, and 0.25, the temporally-averaged sample-mean values are consistently higher than the actual values given at the bottom of the table, but somewhere between ε = 0.25 and 0.5, MMAE appears to lose sensitivity to the partial impairment and hence, the values are much *lower* than the actual value. The only tool at our disposal for improving detection performance of the current MMAE algorithm is retuning, but recall that at the "endpoint" values of ε (0 and 1.0), MMAE appears to be very well tuned. It was for this reason that retuning MMAE to improve performance for partial failures was not considered seriously, except for the rudder, which will be described in Section 4.9. Instead, a decision was made to go forward with the MMAE algorithm tuned as it is and to try to compensate for these inadequacies in other ways.

These results, while somewhat less than desired, were not a hopeless end nor were they a definitive answer on the blending ability of the MMAE algorithm. They *did* require a decision about the best direction

of further research, in order to fulfill the greatest number of thesis goals and to obtain the greatest amount of data in the short time available.

4.5.2. Action Taken Given the Results of Section 4.5.1

It was decided, based on the results of the single actuator, partial impairment tests, to concentrate on three avenues of research. First, it was decided to study the interface problems between MMAE and Control Reconfiguration (CR) posed by the actual characteristics (particularly rapid variations over significant periods vs. constant-in-time as was desired) of blended probabilities resulting from partial actuator impairments. Second, it was thought necessary to try to improve MMAE performance in detecting partial rudder actuator impairments by retuning the MMAE, *without spoiling its existing capability of detecting total impairment of all actuators and sensors.* Third, it was decided to implement and test all additional simulation code necessary for the study of dual, partial actuator impairments with Control Reconfiguration, the final goal being to analyze cases of dual-actuator and actuator-sensor impairments. This final goal was understood to be more of a "feasibility study", not only to exercise and study the present capabilities of the entire MMAE/CR algorithm, but to uncover all possible difficulties and recommendations for future research.

4.6. Interface Problems Between MMAE and Control Reconfiguration

To understand the interface problem between MMAE and Control Reconfiguration during partial impairments, it is helpful to review the method applied by Stepaniak [45] and Lewis [26] and why it was sufficient for their research. Stepaniak examined *single*, total impairments of actuators and sensors, Control Reconfiguration was triggered when the probability value for any actuator passed a threshold value of .95 *for one sample period* (1/64 second), and remained "latched" for the rest of the simulation run of 8 seconds. Furthermore, the Control Reconfiguration (CR) scheme employed a pre-packed "total failure matrix, $\mathbf{D}_{ai}$",

computed for total actuator failures. Since Stepaniak put a great deal of effort into refining his dither scheme and into tuning the MMAE algorithm to detect total actuator impairments quickly and correctly, his method of interface was valid and gave excellent results, since the probability plots displayed nearly ideal behavior. Lewis [26] used the same method for triggering Control Reconfiguration during his *dual*, total impairment study, but added a *leading edge timer* which triggered total-impairment Control Reconfiguration only after the probability threshold exceeded .95 for .156 seconds. This was an attempt to avoid reconfiguring on probability "spikes" which he encountered. MMAE detection of all first total impairments was, like Stepaniak's results, quick and correct and guaranteed proper first-failure-reconfiguration, transition, or "swap" to the correct filter bank, and good detection of the second total actuator impairment. In neither Stepaniak's nor Lewis' work was an online probability estimate from MMAE used directly in actual reconfiguration calculations, but rather it was used as a "one-shot-trigger" to start reconfiguration calculations based on assumptions of totally failed actuators.

This thesis research assumed that MMAE is able to blend appropriate probability channels to obtain an accurate, *stable* estimate of partial actuator impairment. Not only is this estimate used directly in Control Reconfiguration calculations (Eqn. (3-35)), *but it is also used to decide which MMAE filter bank to transition to after the first partial impairment*, since "Level 1" in the MMAE hierarchy (Figure 2-3) now includes three *additional* filter banks (the selection and design of which are covered in Section 4.6.2) for each actuator based on *partial* (and not just total) impairments. Since results of the single-actuator, partial impairment tests presented in Section 4.5.1 demonstrated probabilities with unsteady behavior as well as inaccurate temporally averaged mean values (Table 4-3), the interface problem between MMAE and CR confronted by this thesis becomes clear: *how to extract an accurate, stable estimate quickly from the probability data to use for MMAE filter Bank Switching and for Control Reconfiguration calculations.* It was decided that a two-stage process to smooth and then quantize probabilities *might* be a valid approach to obtain such an estimate.

4.6.1. *Probability Smoothing*

Three methods for smoothing out spikes, jitter, and rough edges in probability time histories were tried: first order filtering, weighted-average filtering, and moving window averaging. The criteria for selecting a good filtering scheme were ease of implementation and conservation of original, temporarly-averaged mean probability values from 1 to 8 seconds, as listed in Table 4-3. An actual probability data file from a single run of the simulation was loaded into Matlab [49], where it was passed to a discretized, first order Butterworth filter with break frequencies from 1 to 3Hz:

$$\frac{p_{filtered}}{p} = \frac{2\pi f}{s + 2\pi f}; f = 3,\ 2,\ 1.5,\ 1.0 Hz \tag{4-5}$$

The 'p' in Equation (4-5) is the probability for one elemental Kalman filter, p_k, and the sample frequency, f_s, equals the simulation update rate of 64 *Hz*. The same probability data file was then passed through a weighted-average filter with weights, α_w, equal to 0.8, 0.85, and 0.9:

$$p_{filtered}(t_i) = \alpha_w p_{filtered}(t_{i-1}) + (1 - \alpha_w) p(t_i) \tag{4-6}$$

In the third approach, the data was averaged using a moving window scheme and trying 3 different window lengths, 'N':

$$p_{filtered}(t_i) = \frac{1}{N} \sum_{j=i-N+1}^{i} p(t_j) \tag{4-7}$$
$$N = 10,\ 20,\ 30 \text{ samples}$$

The original probability data and the results of all three approaches are shown in Figures 4-4 a), b), and c):

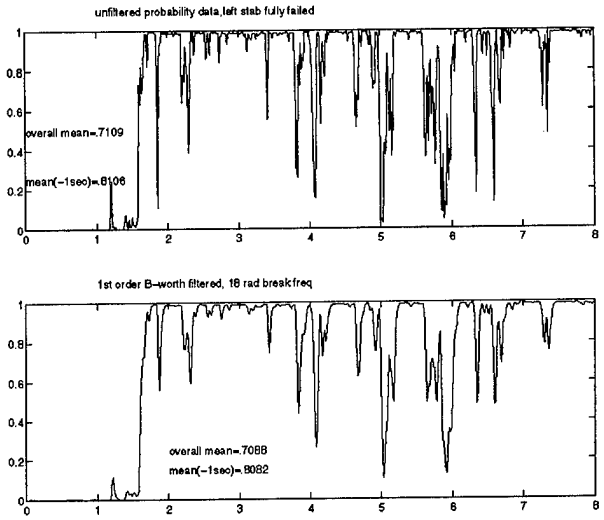


Figure 4-4 a) Raw and Low-Pass Filtered Probability Data for Left Stabilator Failure

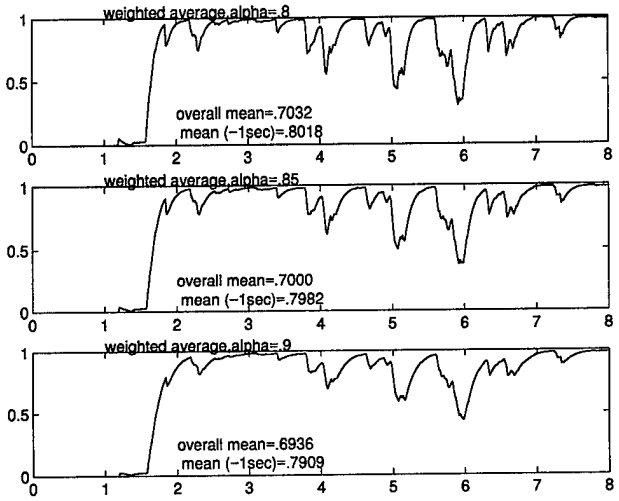


Figure 4-4 b) Probability Data Filtered Using Weighted Average Filtering

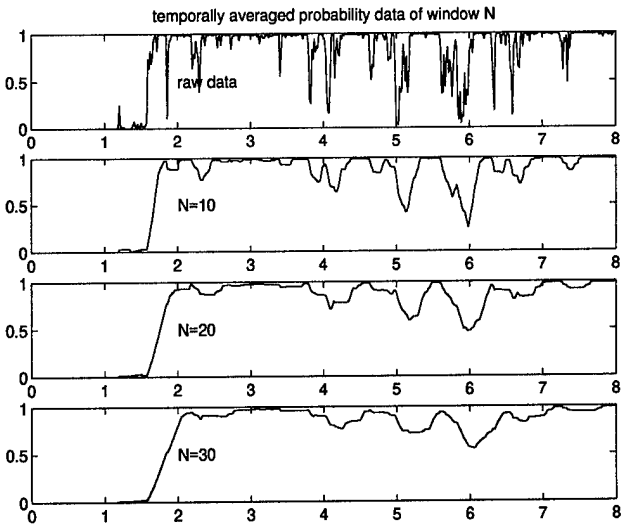


Figure 4-4 c) Probability Data Filtered Using Moving Window Averaging

Using the criteria mentioned above, the weighted-average filter with α_w=.8 was chosen and implemented in the VISTA F-16 truth model simulation.

4.6.2. Discrete Partial Failure Levels and "Probability Quantization"

The need to "quantize" the filtered probabilities to discrete levels was apparent because the filtered probabilities still displayed "drop-outs" and oscillatory behavior about the mean value (Table 4-3). Being rapidly and constantly varying, they were still not suitable for direct input into Control Reconfiguration, since it is undesirable to reconfigure continuously and possibly drive the actuators into rate and/or position limits. It was also necessary to select the discrete levels of possible partial failures somehow when considering future design of the necessary "partial fail" filter banks for MMAE. Recall that, for partial failures of single actuators, where MMAE is not required to transition (or "swap") from Level '0' to Level '1' (Figure 2-3), MMAE computes an estimate of the partial impairment by blending the outputs of two filters which exist already, namely the "fully-functional aircraft" (ff) Kalman filter and the appropriate fully-failed Kalman filter corresponding to the impaired actuator. For dual impairments, where MMAE is required to transition to the next level of Kalman filters, there is no path in the *current*, hierarchical, moving-bank MMAE algorithm (Figure 2-3) between any *partial first-impairment* (taking place at Level

'0') and a second impairment at Level '1'. To provide that path, it is necessary to add "partial-fail" filter banks to Level '1' for each of the five actuators, as well as the "switching mechanism" to get to the new banks. An example of the partial fail filter bank for the left stabilator is illustrated in Figure 4-5, where three partial fail filter banks have been constructed and are shown in addition to the "total failure" bank that existed already. Since the number of partial-fail filter banks cannot be infinite, we must reduce the infinite number of partial failures possible between 0 and 1 to a manageable number of discrete, or "quantized" impairment levels. It is seen in Figure 4-5 that each level selected requires the construction of 11 new Kalman filters, hence it is up to the designer to select the minimum number of quantized, partial failure levels and their values.

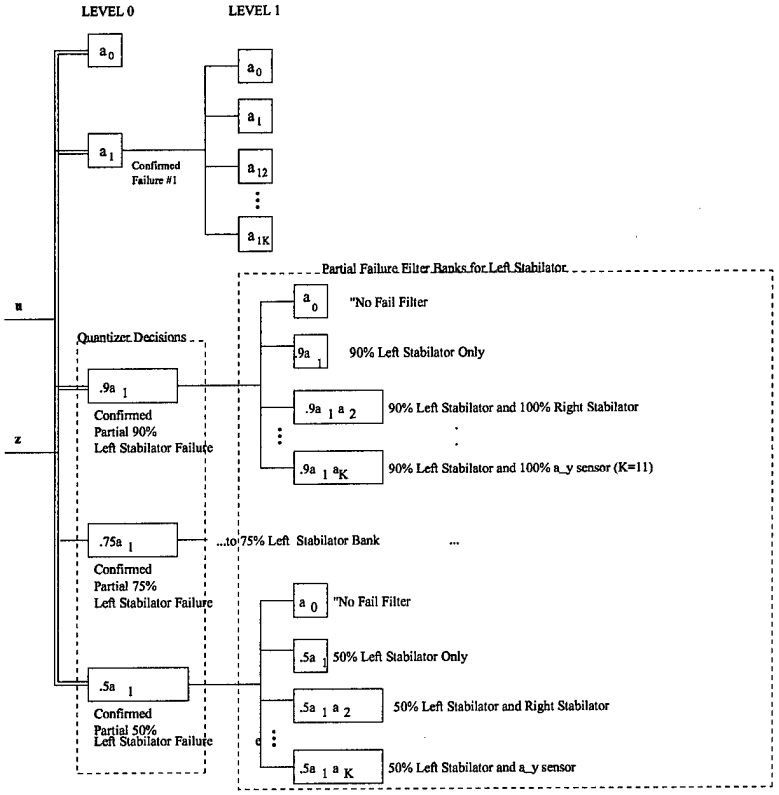


Figure 4-5 Partial Failure Filter Banks for Left Stabilator

As Figure 4-5 indicates, three discrete, partial failure levels were chosen for each actuator, based upon trends observed visually in the Probability Summary Plots and upon the results of Table 4-3. The values of these levels were, for this thesis, chosen to be: 0.9, corresponding to a 90% actuator impairment (i.e. $\varepsilon = 0.1$, or 10% actuator effectiveness), 0.75, corresponding to a 75% impairment, and 0.5, corresponding to a 50%

partial actuator impairment. Since MMAE appeared to be insensitive to partial impairments below 50%, no values less than 50% were considered. The partial impairment discrete levels now defined, the additional banks of Kalman filters necessary for partial impairment detection, and the expansion of the filter bank switching algorithm necessary to transition to these banks, could be designed, constructed and implemented in the VISTA F-16 simulation. It was still necessary, however, to provide the vital link between the constantly varying probability signals and the relatively constant probability estimate upon which to base bank switching and Control Reconfiguration. The method employed to accomplish this was the construction of a "Probability Quantizer", described in the next section.

4.6.3. Probability Quantizer

The probability quantization algorithm examines the current MMAE probability value, p_k, for each of the five actuators and assigns values of quantized probability, $p_{k_{quantized}}$ based on a minimum amount of time spent at or above the *highest* of four possible threshold values: 0.95, 0.9, 0.75, and 0.5. Because the probabilities never actually attain a "perfect" value of 1.0 (for reasons explained earlier), the threshold value of 0.95 is, like in earlier theses [26, 45], necessary as a practical boundary above which to declare total actuator failure. The next three threshold values were chosen, for the first design, to be equal (i.e. with no built-in "bias") to the discrete, partial failure levels which were assumed in the design of the partial failure filter banks. There were no threshold levels below the value of 0.5, so any value less than 0.5 was assigned the value of zero, given the current insensitivity of MMAE to detection of partial impairments of less than 50%. "Leading and trailing edge timers" are employed on all four thresholds within each channel to ignore brief probability "spikes" and "drop-outs", by tracking the time that the incoming filtered probability data has been at or above that threshold level, and setting the threshold 'high' if the time is *greater than* or equal to the preset timer value (currently .15 seconds). Likewise, trailing edge timers keep the threshold level 'high' until the incoming probability data has been *below* that threshold value for a predetermined amount of time (currently .5 seconds). An example of such a quantized output given filtered probability input data for the left stabilator is shown below in Figure 4-6. At each instant of time, the

resultant, quantized probability for any one of five actuator channels is the current highest threshold level that is non-zero. All parameters of the Probability Quantizer Algorithm, including threshold levels as well as leading and trailing edge time constants, are "tuning parameters" and are adjustable by the user. At the current time, all leading edge time constants are set to 0.15 seconds and trailing edge constants are set to 0.5 seconds. These values were chosen to ensure a rapid responsive by the quantizer to the initial detection, but to retard any dropouts in filtered probability.

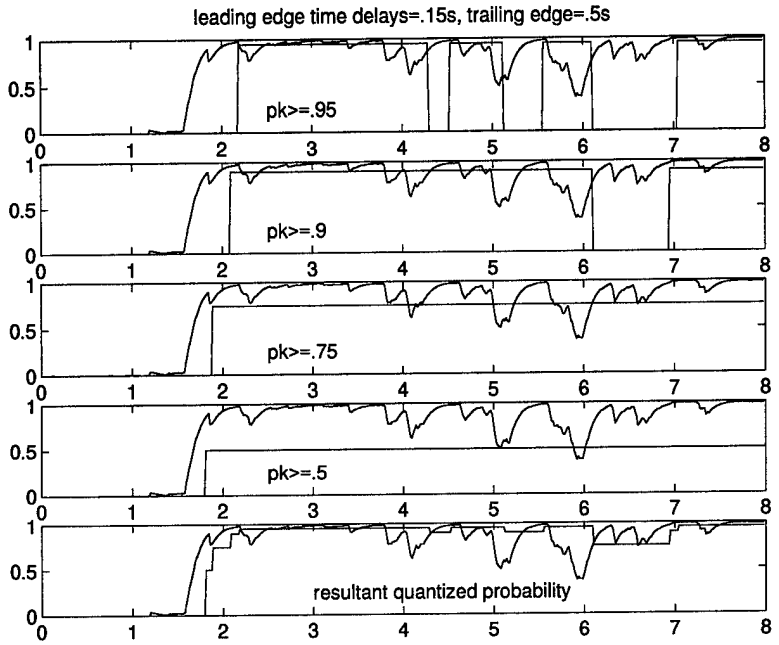


Figure 4-6 Example of Quantized Probability

4.6.4. Purposeful Biasing

The Probability Quantizer is *the* interface between raw probabilities and the bank swapping logic within MMAE, as well as between MMAE and Control Reconfiguration. The bank swapping logic selects a "Level 1", "partial-fail" or "total-fail" filter bank based upon the quantized probability value that it "sees" for each actuator channel. As currently implemented, there are no "effective biases" between any quantized probability values and the partial impairment filter banks selected by the bank swapping logic: if a quantized probability value is 0.9, for example, and MMAE has yet not switched to any "Level 1" filter bank, then the bank swapping logic selects the partial-failure bank corresponding to a 90% actuator

impairment.) Furthermore, that quantized probability value is sent simultaneously to the Control Reconfiguration module as the actual estimate of impairment. It is an easy task, if necessary, to "bias" any or all estimates by mapping, or "calibrating" a known impairment level to an observed quantized probability value, simply by changing the logic in the bank swapping algorithm to swap to a different filter bank for an observed value of quantized probability. Thus, if performance evaluations indicate, for example, that a quantized probability value of 0.65 on a left stabilator failure corresponded to an actual 75% impairment level, then such a purposeful bias could be incorporated into the declared level of impairment to be used by the bank switching logic and Control Reconfiguration. (Let the reader be warned that, if this sort of "purposeful biasing" is implemented, then it is *mandatory* that the quantized probability estimate currently being sent to Control Reconfiguration must be biased as well, or replaced by the assumed, and constant-with-time, impairment level of the filter bank).

4.7. *Design of Partial Impairment Kalman Filter Banks*

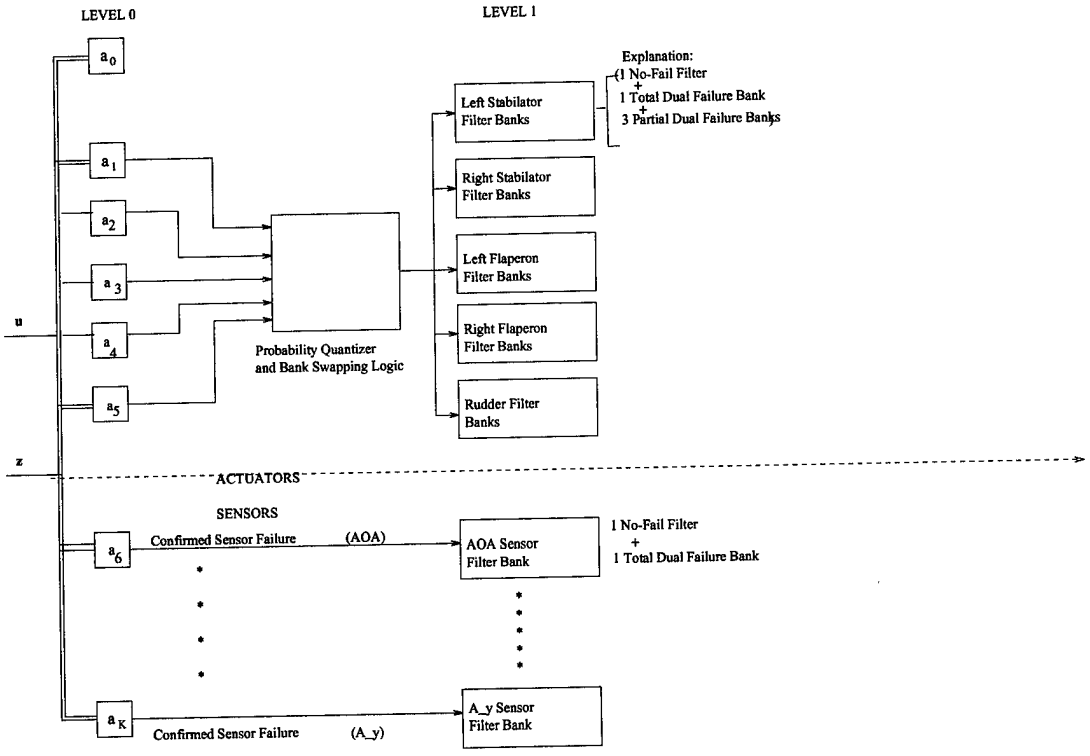


Figure 4-7 Expanded MMAE Hierarchy

The next task was to design and construct the additional banks of Kalman filters needed to detect 15 possible initial, partial actuator impairments. Recalling from Section 4.6.2 that three possible discrete levels of partial impairment, 0.9, 0.75, and 0.5, were chosen for each of 5 actuators, the task then became the construction and integration of 15 more banks of "Level 1" Kalman filters added to the existing 12-bank MMAE structure (Figure 2-3). Necessary modifications were made to Stepaniak's original algorithm for constructing these filter banks. Figure 4-7 shows a top-level view of the new, expanded 27-bank MMAE bank structure which resulted when these 15 "partial-failure" filter banks were added. It is very important to realize from Figure 4-7 that transition to the correct, "Level 1" actuator failure filter bank is totally dependent on the ability of MMAE, specifically, the probability quantizer, to estimate the first actuator impairment correctly, and then pass the estimate to the bank swapping logic; i.e. *there are no Kalman filters in "Level 0" constructed and tuned specifically to a single, partial failure hypothesis.* The implications of this observation are made clear in the next section.

Bank Switching Algorithm

For reasons already stated, the logic and methods used by earlier theses [26, 45] to transition between filter banks within the MMAE algorithm were not sufficient for this research and had to be modified extensively. Switching from "Level 0" to "Level 1" MMAE filter banks now depends solely upon quantized probability values of "Level 0" filters. Prior to detection of the first actuator impairment, the bank switching algorithm examines quantized probability values for all actuators and identifies the actuator with the largest value at or above .5, and then declares the first failure (partial or total) and switches to the appropriate "Level 1" bank. If an improper bank swap is made by the algorithm, then the MMAE will, most likely, remain in that filter bank. The only method MMAE has, at this time, to *back-out* of a bank is if the "no-fail" filter (filter "a_0" in Figure 4-7) has probability activity substantial enough to trigger a transition out of the bank and back to "Level 0". This is an unlikely occurrence, unless the bank switching algorithm is made highly sensitive to any non-zero probability activity on the "no-fail" filter channel. The means of detecting sensor total failures (no partial failures were studied in this thesis) remains unchanged from earlier

thesis efforts. If a sensor failure probability surpasses a trigger threshold of 0.95 for 1 sample period ($1/64^{th}$ of a second), then a sensor total failure is declared. If the sensor failure is the first impairment, or if it is merely detected first before an actuator impairment, then the bank switching algorithm will execute a filter bank swap to a "Level 1" filter bank hypothesizing a total, first impairment of a sensor. In other words, when describing the MMAE bank switching algorithm, it is "first-detected, first-swapped".

4.8. Dual Impairment Control Reconfiguration

Control Reconfiguration takes place immediately after filter bank switching occurs, using the time-varying, quantized probability estimate and identification number (1 through 5) of the impaired actuator. Control reconfiguration calculations do not, at this time, use the *implicit, fixed value* of partial impairment assumed in the design of the currently active filter bank. Initially, this design decision appeared to make the most sense, for it was thought that it would yield the better-performing Control Reconfiguration solution. In lieu of actual MMAE blending performance, and the compensation techniques which may be used (biasing) to improve performance, this should, perhaps, be changed in the future to the *implicit* impairment estimate.

Control Reconfiguration occurs in one or two "stages", depending on whether there are one or two actuator impairments. "Stage 1" reads the identification number and quantized probability value of the first impaired actuator and either 1) reconfigures for a total actuator impairment using the pre-packed $\mathbf{D}_{ai}$ matrix (Equation (3-37)) for that actuator, or 2) reconfigures for a partial impairment using equivalent but simplified calculations to approximate pseudoinverse, or in partial impairment cases, true inverse calculations. If there is a second impairment, the reconfigured actuator commands from Stage 1 are passed along with information on the second actuator impairment to "Stage 2", where the second reconfiguration is carried out in a manner identical to Stage 1.

4.9. *Retuning the MMAE Rudder Channel*

Concurrent with the extensive algorithm developments and enhancements outlined in Sections 4.6 through 4.8, it was necessary to try some technique to improve MMAE ability to detect partial rudder impairments. As we observed in Section 4.5, MMAE declared the rudder to be totally impaired for cases where the actual impairment ranged from total (100%) to 50%. Somewhere in the range of partial impairments between 50% and 75%, the "fully functional" (ff) and "rudder failed" (rud) probabilities start to blend in a reasonable manner, which is evident by comparing the rudder (rud) channels of Probability Summary Plots for the cases of 50% and 75% actuator impairments, with no Control Reconfiguration, found in Figures 4-13 and 4-11 later in this chapter. This behavior presented a problem. Since the MMAE algorithm was apparently very well tuned at the "endpoints" to detect total rudder actuator failure, or to indicate fully functional behavior without false alarms when no impairments were inserted, the task was to improve "mid-impairment range" performance without destroying exceptional MMAE performance at the endpoints. Two different methods of tuning the rudder channel within MMAE were attempted: depositing additional amounts of *discrete-time* white dynamics pseudonoise (i.e., increasing the magnitude of a component of $\mathbf{Q_d}$ as given in Eqn. (3-25)), onto the rudder actuator *position*, and adding amounts of *continuous-time* white dynamics noise (as incorporated into Eqn. (3-22) *before* discretization), onto the rudder *state*. Values of discrete-time measurement noise covariance, $\mathbf{R}(t_i)$ were left unchanged from their values given in Table 3-9.

4.9.1. *Discrete-Time, White Dynamics Pseudonoise*

Discrete-time noise added onto the rudder actuator *position* is equivalent to adjusting the covariance of $\mathbf{w}'_d(t_i)$ in:

$$\mathbf{x}_{aug}(t_{i+1}) = \Phi\,\mathbf{x}_{\mathbf{aug}}(t_i) + \mathbf{B_d}\mathbf{u}_{\mathrm{aug}}(t_i) + \mathbf{w}_d(t_i) + \mathbf{w}'_d(t_i) \tag{4-8}$$

A self-contained program written by Stepaniak [45] especially for tuning MMAE in this manner was used to explore the effect of adding increasing amounts of discrete-time, white noise onto the rudder actuator position (state 13) of all Kalman filters within the MMAE structure. When given a starting value, an increment, and an ending value of noise variance, q_d, the program first builds all Kalman filters, adding the current value of q_d after filter discretization and immediately before computation of the (assumed) constant Kalman filter gains $\mathbf{K}$ and matrix $\mathbf{A}_k^{-1}$ in Equations (2-6) and (2-7). The automated VISTA F-16 simulation is run next, examining all single, total or partial actuator impairments. A Probability Summary Plot is then printed, the value of q_d is incremented, and the process is repeated until the upper bound on q_d is reached. The process just described was repeated for a 50% rudder impairment and q_d values of: 10^{-9}, 10^{-7}, 10^{-5}, and 10^{-3}. Unacceptable degradation in detection performance was observed in the MMAE "fully functional" (ff) channel and in actuator channels other than the rudder, while the rudder channel still indicated a tendency to declare a fully failed rudder, but now with frequent probability "drop outs" to zero. As a result of these disappointing trends, this tuning experiment was abandoned.

4.9.2. *Continuous-Time White Dynamics Pseudonoise on States*

This is accomplished via:

$$\dot{\mathbf{x}}_{aug}(t) = \begin{bmatrix} \mathbf{A} & \mathbf{B}_{\mathrm{mod}} \\ \mathbf{0} & -14\cdot\mathbf{I} \end{bmatrix} \mathbf{x}_{\mathbf{aug}}(t) + \begin{bmatrix} \mathbf{0} \\ 14\cdot\mathbf{I} \end{bmatrix} \mathbf{F}_{\mathbf{ai}}\mathbf{u}_{\mathrm{aug}}(t) + \begin{bmatrix} \mathbf{G} \\ \mathbf{0} \end{bmatrix} \mathbf{w}(t) + \begin{bmatrix} \mathbf{B}_{actcol\,8x1} \\ \mathbf{0}_{5x1} \end{bmatrix} w_b(t) \tag{4-9}$$

In this approach to tuning, suggested by Eide [14], a scalar pseudonoise, $w_b(t)$, of adjustable strength q is brought into the propagation equations via multiplication by $\mathbf{B}_{actcol}$, where $\mathbf{B}_{actcol}$ is the 8-by-1, 5th column of $\mathbf{B}_{\mathbf{mod}}$ (Equation (3-6)). It can be seen in Equation (4-9) above, that the first 8 state derivatives are affected immediately by the addition of pseudonoise in this approach, while the previous approach added uncertainty only to the rudder position. The simulation used for adding discrete-time noise, specifically, the Kalman filter generation routine, was modified to implement this second method of tuning.

Four different strengths of q were tested: 10^{-3}, 10^{-2}, 10^{-1}, and 10^{0}, for a 50% partial rudder actuator impairment. Results were as disappointing as in the prior tuning method, in that MMAE detection performance for actuators other than the rudder was destroyed well before any beneficial effects were seen in the MMAE rudder channel.

Other methods of improving MMAE detection of partial rudder actuator impairments were explored. The amount of discrete-time measurement noise on the yaw rate sensor was increased slightly to examine the effects, which were, once again, unacceptable degradation in other actuator and sensor channels for useful change in the rudder actuator channel. The frequency of measurements was decreased next to observe what the effects might be of letting the Kalman filter residuals "grow" between measurements. The necessary parameters were set in the VISTA F-16 simulation to decrease the measurement frequency from 64 to 16 *Hz*, thus letting the differences in outputs (due to differing hypotheses) grow for 3 time steps between measurements. The result of doing this unfortunately was to miss detection of any kind of rudder impairment. The reasons for this result were not explored, for in the interest of time, it was decided to move forward to other demanding tasks, noting that special ad-hoc methods might have to be employed for detection of partial impairments of the rudder actuator. One such method might include preliminary *declaration* of a 50% rudder failure for a probability-indicated 100% failure, then further analysis of residual components to resolve the failure further.

4.10. Dual Impairment Analyses

The three sections which follow, Sections *4.11* through *4.13*, examine the results of impairment tests with the entire revised and expanded MMAE / Control Reconfiguration (MMAE/CR) algorithm, as described in preceding sections, "in the loop". Each section begins by presenting and analyzing data for cases of single, total or partial impairments, with bank swapping disabled, first without, and then *with* Control Reconfiguration. This is a logical progression which will lead to the analysis and critique of the dual impairment cases, with the entire MMAE/CR algorithm "up and running", which follow immediately in each of the three sections of data. A critique of Control Reconfiguration performance taken by itself will

be offered where appropriate in each section, but it should be remembered that the purpose of this thesis research was primarily to examine the ability of MMAE to detect and then to provide an accurate, partial impairment estimate to supply to Control Reconfiguration in a timely manner. Prior theses have demonstrated conclusively that, once supplied with the proper identification of the impaired actuator or sensor and with an accurate estimate of the impairment, Control Reconfiguration does indeed benefit performance. All impairment cases are performed without aircraft maneuvering, since the MMAE algorithm does not currently have any sort of gain scheduling to compensate dynamically for any maneuvering, and using Stepaniak's [45] control dither sequence, presented for review in Table 4-4, for enhancement of failure identification. First impairments of actuators are inserted into the F-16 truth model at 1 second, followed by the second impairment at 2 seconds, and all cases presented are the result of 10 Monte Carlo simulation runs.

Aircraft Axis	Control Input	Command Magnitude (lbs)	Dither Frequency (rad/sec)	Phase Added to Dither Signal (degrees)	Actual Command to F-16 Flight Control System
Longitudinal	Pitch Stick	+12/-12.4	7.5	90	Normal Acceleration
Lateral	Roll Stick	11	15	0	Roll Rate
Directional	Rudder Pedals	30	15	0	Sideforce

Table 4-4 Stepaniak Dither Signal

4.11. MMAE/CR Performance During Single and Dual, Total Impairments

This section begins by presenting results of MMAE/CR performance during *single, total* (100%) failures of actuators and sensors *without* Control Reconfiguration or MMAE filter bank swapping, followed by the same impairment cases *with* Control Reconfiguration. Comparisons are made between the reconfigured and non-reconfigured cases (with filter bank swapping 'off')to alert the reader to differences in MMAE performance which exist *solely as a result of Control Reconfiguration. Dual, total* impairment cases *with* Control Reconfiguration and bank swapping enabled are presented next and are compared in a tabular format to the single "100% impairment" results with reconfiguration just described. Since these cases will also include cases where there is no second impairment, it will be possible to assess MMAE

performance for single impairments with Control Reconfiguration *and* filter bank swapping. It is hoped that the presentation of results in such an incremental fashion will allow the reader not only to evaluate overall MMAE/CR algorithm performance, but also to compare and to estimate any MMAE performance differences resulting from the sequential addition of Control Reconfiguration and bank swapping.

4.11.1. Single, Total Actuator Impairments, No Control Reconfiguration

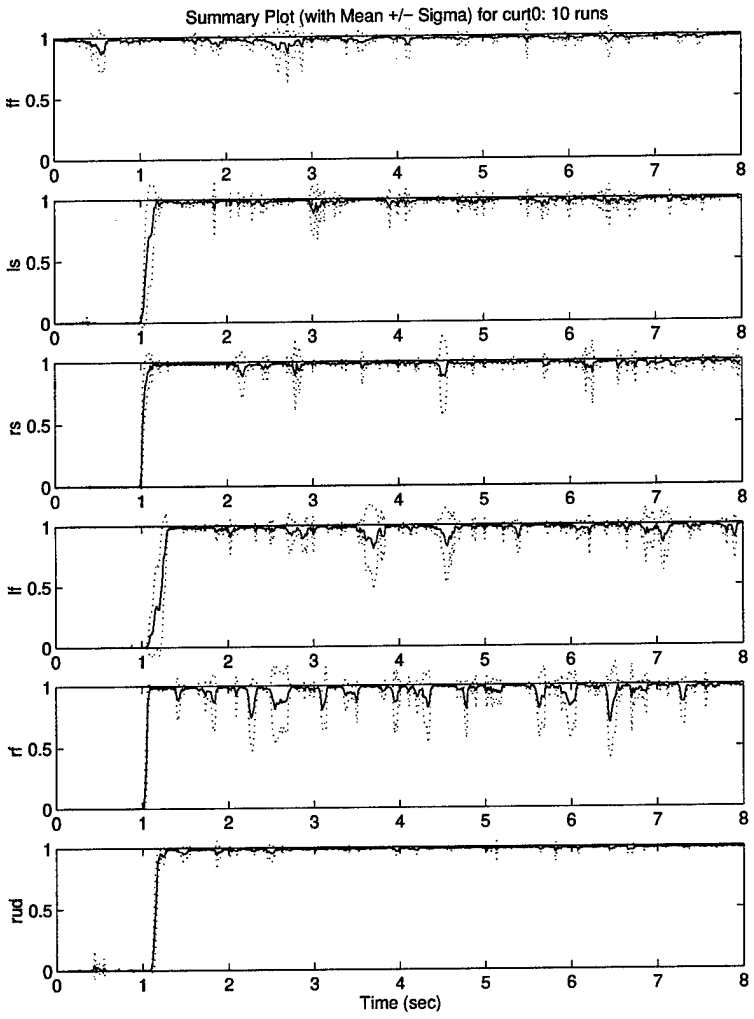


Figure 4-8 Probability Summary Plot for *Single, Total* Actuator and Sensor Impairments *Without* Control Reconfiguration And With Filter Bank Swapping 'OFF"

Figures 4-8 and 4-9 show the Probability Summary Plots when single, 100% actuator and sensor impairments are inserted at 1 second into the simulation. Each Probability Summary Plot displays the "Mean Probability $\pm$ 1 Standard Deviation" of the probability of each filter, $p_k(t_i)$, in the currently active filter bank within MMAE resulting from 10 Monte Carlo runs of the simulation, which runs for 8 seconds. Figure 4-8 displays the results without Control Reconfiguration or MMAE filter bank swapping activated. It is seen that these results are identical to those of Stepaniak [45]. They display the same excellent baseline MMAE performance characteristics of correct and very rapid detection with very little probability spillage or false alarms onto other MMAE channels. Furthermore, these single detections are followed by excellent "lock and hold" performance, with no probability "drop outs" onto either the "fully functional" (ff) or any other MMAE channels.

4.11.2. Single, Total Impairments With Control Reconfiguration (Bank Swapping 'Off')

Figure 4-9 displays results of single, 100% impairments when MMAE filter bank swapping is left disabled, but Control Reconfiguration is activated. These results show performance different from the cases without Control Reconfiguration. Detection performance in Figures 4-8 and 4-9 is identical when examining the leading edge shapes of the probability plots, but probability "lock and hold" performance has deteriorated, especially for the rudder impairment case (rud), which exhibits frequent probability drop-outs and losses starting at about 1.8 seconds into the simulation. Examining the corresponding Probability Plot (Appendix A.2) for this case, it is seen that spillage occurs most noticeably onto the yaw rate sensor channel, and to lesser degrees onto all other actuator and sensor channels. Figure 4-9 also shows that the "Right Stabilator" (rs) and "Right Flaperon" (rf) channels also exhibit periodic probability drop-outs during the lock and hold phase, but to a much lesser degree than in the rudder case. The probability drop-outs in these cases appear to coincide to aircraft "state reversals", when states, such as sideslip, β, and bank angle,

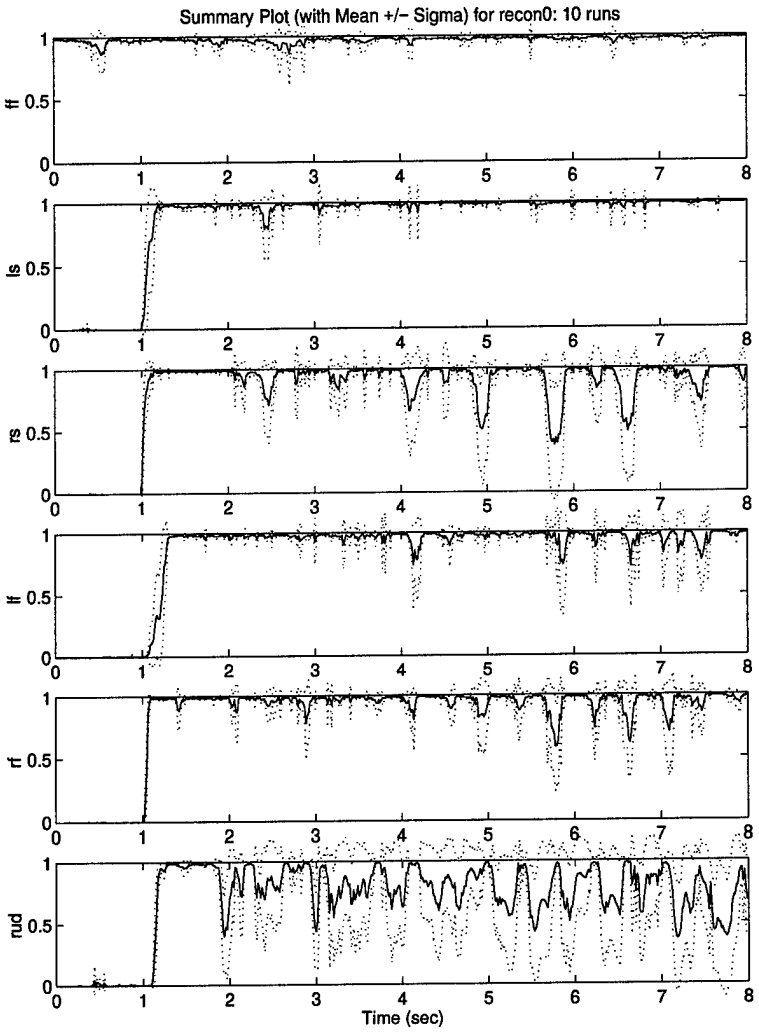


Figure 4-9 Probability Summary Plot for *Single, Total* Actuator and Sensor Impairments *With* Control Reconfiguration (Bank Swapping 'OFF')

ϕ, change direction. Though exact causes could be found given sufficient additional time, one plausible explanation is that this may be when maximum error occurs between the VISTA F-16 truth model and the Kalman filter-based linear models, due possibly to unmodeled, higher order effects. Tables 4-5, 4-6, and 4-7 together provide a summary comparison of the impairment cases just discussed. Table 4-6 was prepared by examining the Probability Summary Plots listed above along with the appropriate Probability Plots in Appendices A.1 and A.2. The top row of Table 4-6 lists all 12 MMAE actuator and sensor channels, including the "fully functional" (FF) channel. The left-most column is divided into 2 rows, labeled "***No*** C/R" (No Control Reconfiguration), and "C/R" (Control Reconfiguration activated). Entries in the table

ranging from "Good" to "No Detect" (ND) rate the quality of MMAE performance and the ratings are explained further in Table 4-7.

MMAE Channel Abbreviation	**Fully Failed Filter Hypothesis (Actuators)**	**MMAE Channel Abbreviation**	**Fully Failed Filter Hypothesis (Sensors)**
FF	Fully Functional (**No Failure**)	AOA	Angle-of-Attack
LS	Left Stabilator	Q	Pitch Rate
RS	Right Stabilator	A_n	Normal Acceleration
LF	Left Flaperon	P	Roll Rate
RF	Right Flaperon	R	Yaw Rate
RUD	Rudder	A_y	Lateral Acceleration

Table 4-5 MMAE Probability Channel Abbreviations and Corresponding Filter Hypothesis

	FF	LS	RS	LF	RF	RUD	AOA	Q	A_n	P	R	A_y
No C/R	Good	Good	Good	Good	Good	Good	Good	Good	Good	Good	Good	Good
C/R	Good	Fair	Fair	Good	Good	Poor	Good	Good	Good	Good	Good	Good

Table 4-6 Summary Comparison of Single, Total Impairments *with* and *without* Control Reconfiguration (Bank Swapping 'OFF')

Rating	Qualitative Description of Ratings
Good	Rapid detection on correct MMAE channel. Good probability lock and hold. Trace of probability "spillage" but not persistent. Good signal for Control Reconfiguration
Fair	Good rating except now significant dropouts from probability lock. Could still get good signal for Control Reconfiguration.
Poor	Performance drastically degraded from case without Control Reconfiguration because of frequent spikes and drops in probability with no lock. Accurate signal for Control Reconfiguration not possible. Also marked by excessive "spillage" onto other channels
ND	No detection, or an incorrect MMAE detection.

Table 4-7 Explanation of Ratings in Table 4-6

Note that all cases *without* Control Reconfiguration (CR) are good, but that there are differences in ratings when CR is activated. Specifically, the rudder (RUD) case goes from "Good" to "Poor", and both stabilators (LS, RS) are given "Fair" ratings. The rudder case in Figure 4-9 exhibits strongly periodic dropouts in probability, starting at about 4 seconds. A review of the corresponding Probability Plots in Appendix A.2 reveals that very noticeable probability "spillage" occurs on the yaw rate sensor (R), and normal acceleration and pitch rate channels (A_n, Q). This is possibly due to the control dithering signals being "scrambled" by Control Reconfiguration, in the sense that the phasing and relative amplitudes of those signals are no longer the same. The left stabilator (LS) channel exhibits a very slight degradation in

lock-and-hold performance, giving up a momentary probability "spike" to the "fully functional" (FF) channel, while the right stabilator (RS) gives up more significant, periodic "humps" to the (FF) channel. Both are given "Fair" ratings because of these performance degradations.

Since all dual impairment simulations which follow include one simulation (consisting of 10 MC runs) of *only* the first impairment and no second impairment, a special simulation "batch run" consisting of *single actuator impairments with CR and bank swapping* was not performed, since the data is gathered during the following Dual Impairment simulations and is available by looking at the Probability Plots in the Dual Impairment Appendices: D.1 through D.3.

4.11.3. Dual, Total Impairments with Control Reconfiguration and Bank Swapping

Results of MMAE performance during single, 100% actuator and sensor impairments and the differences in MMAE performance due to Control Reconfiguration have just been examined. We now compare those results with the following results of dual, total impairment cases with the full MMAE algorithm "up and running", to include filter bank swapping and Control Reconfiguration. This comparison is presented in Table 4-8 using the "comparison matrix" and rating criteria for dual, total failures established by Eide [14] and Lewis [26]. The left-most column of Table 4-8 represents the first total (100%) impairment of an actuator inserted at 1 second, while the upper-most row of Table 4-8 represents the second impairment occurring at 2 seconds. This time spacing between actuator impairment events was chosen so that the second impairment occurred before the aircraft had a chance to venture away from the design point (Mach .4, 20000') of the MMAE algorithm. Also, there are no cases in which sensors are the first impairment in Table 4-8, since those cases were not performed in the interest of time. Entries in the table ranging from "Good" to "No Detect" (ND), explained further in Table 4-9, rate the quality of MMAE detection and "lock and hold" performance during the second impairment versus the quality of MMAE performance seen for single impairment cases. For example, a rating of "Good" in the table entry corresponding to a first impairment of "left stabilator" and second impairment of "right stabilator", (LS,RS) = "Good", means that "when a left stabilator is the first total impairment, a right stabilator total

impairment is detected by MMAE with quality as good as if the right stabilator were the only impairment". This type of relative comparison, then, gives the reader an idea of MMAE performance degradation in detecting second impairments. **BOLDFACE** entries on the diagonals show MMAE single detection performance when Control Reconfiguration and filter bank swapping are activated but when there is no second impairment. This is an extremely important rating, since deterioration in this rating will mean that the MMAE is not performing correctly due to [improper] bank swapping. Two modifications are made to this table, namely, the use of an asterisk (*) and the use of a second rating underneath the first and separated by a slash "/". The asterisk is a qualifier appended usually to "Poor" ratings, and is an attempt to enhance the rating by letting the reader know that, in spite of the low rating, the overall "1st detection / 2nd detection profile" showed "good tendencies", or a *definite potential* for better performance. The use of a second rating below the first, or primary rating, is an attempt to judge *steady state*, or MMAE "lock and hold" performance by a different criterion, and hence to "upgrade the rating" being given. It could be argued that *perhaps* MMAE lock and hold performance should be judged for only a given time period after the second impairment, the rationale being that by, say, 2 seconds after the second impairment, that impairment will have been identified and compensated by Control Reconfiguration. If the "2 second" criterion is appropriate, then some of the performance ratings in the table should be given the better rating listed underneath the first.

Second Impairment

	LS	RS	LF	RF	RUD	AOA	Q	A_n	P	R	A_y
LS	**FAIR/ GOOD**	Good	Good	Good	Fair/ Good	Fair	Good	Good	Good	Good	Good
RS	Fair	**FAIR/ GOOD**	Good	Fair	Good	Fair/ Good	Good	Fair/ Good	Fair/ Good	Fair/ Good	Good
LF	Good	Fair	**GOOD**	Poor*	Poor*	Poor*	Poor*	Poor*	Poor*	Poor*	Poor*/ Fair
RF	Fair	Good	Fair	**GOOD**	Fair	Fair	Fair	Fair	Fair	Poor*/ Fair	Fair
RUD	Fair*	Fair*	Poor	Poor	**POOR**	Poor*	Poor*	Poor*	Poor*	Poor*	Fair*

Table 4-8 Dual, Total Impairment Evaluation

Rating	Qualitative Description of Ratings for Dual, Total Impairments
Good	Second detection is on correct MMAE channel and shows performance equal to or better than first. Good Probability Lock and Hold after second impairment. Good estimate for Control Redistribution
Fair	Second impairment displays performance degraded from single impairment case. Temporary but significant dropouts from probability lock. Could get fair estimate for Control Reconfiguration.
Poor *	Performance drastically degraded from single failure case because of frequent spikes and drops in probability with no lock. Accurate signal for Control Reconfiguration unlikely. Also marked by excessive "spillage" onto other channels or "indecision" by MMAE on a sample by sample basis Asterisk (*) means that trends/probability profiles were very promising in spite of rating given
ND	No detection, or an incorrect MMAE detection for second impairment
BOLDFACE	Rates MMAE ability to continue detecting and estimating first partial impairment when no second impairment is inserted, but filter bank swapping has occurred.

Table 4-9 Rating Definitions for Table 4-8

The first observation we should make from Table 4-8 is that the boldface diagonal terms, corresponding to cases of single stabilator and rudder impairments *with Control Reconfiguration and filter bank swapping*, show a degradation in MMAE performance. A review of the Probability Plots for these cases in Appendix D.1 reveals that the left stabilator is rated "Fair" because of momentary but significant probability dropouts (given to the rudder channel) occurring past 5 seconds. If the "2 second" rating criterion applies, then the rating is upgraded to "Good", since a visual cross check between the Probability Plot and Figure 4-9 reveals identical probability traces for the first few seconds. (In this case and in the ones which follow, research time remaining prevented an in-depth analysis of the reasons for the differences observed.) The right stabilator (RS) suffered the same problems of probability dropout as the left stabilator, but in this case gave up probability to the "fully functional" (FF) MMAE channel. It too should be considered "Good" if the alternate criterion is used. The rudder (RUD) channel showed an initial, "solid" or decisive total detection, but at 2 seconds started suffering massive probability dropouts to all channels and in particular to the sensors; hence, it was rated as "Poor" for the reasons given in Table 4-9. The cases of left flaperon total impairment followed by second total impairment of the right flaperon or rudder bear special attention. Probability Plots for those cases reveal a "clipped" appearance on the probability traces, for the correct two actuators, after the second impairment at 2 seconds. In the first case (LF/RF), both probability traces are clipped to a value of exactly .5, which indicates that out of 10 Monte Carlo simulation runs, the dual, total impairments were detected *correctly* by MMAE for *5 of the10 Monte Carlo runs.* (This is not the same as saying "for half of the time", which would *mean within the same MC run*, and would result in smooth probability dropouts, or "trade-offs" between channels.) In the remaining 5 Monte Carlo runs of the "left

flaperon/right flaperon" impairment scenario, only the left flaperon was detected (with some spillage onto left stabilator and yaw rate sensor channels) and the right flaperon showed an initial detection, but then the probability dropped to zero. A similar analysis applies to the case of total "left flaperon/rudder" impairment, the slight difference being that the ratio of Monte Carlo runs with the correct:incorrect detections was 6:4. Control Reconfiguration, and not bank swapping, has caused the indecision by MMAE in this case, and most likely in the cases which follow where probability "clipping" is present. This was apparent when this particular case was rerun with Control Reconfiguration turned off (Figure 4-10).

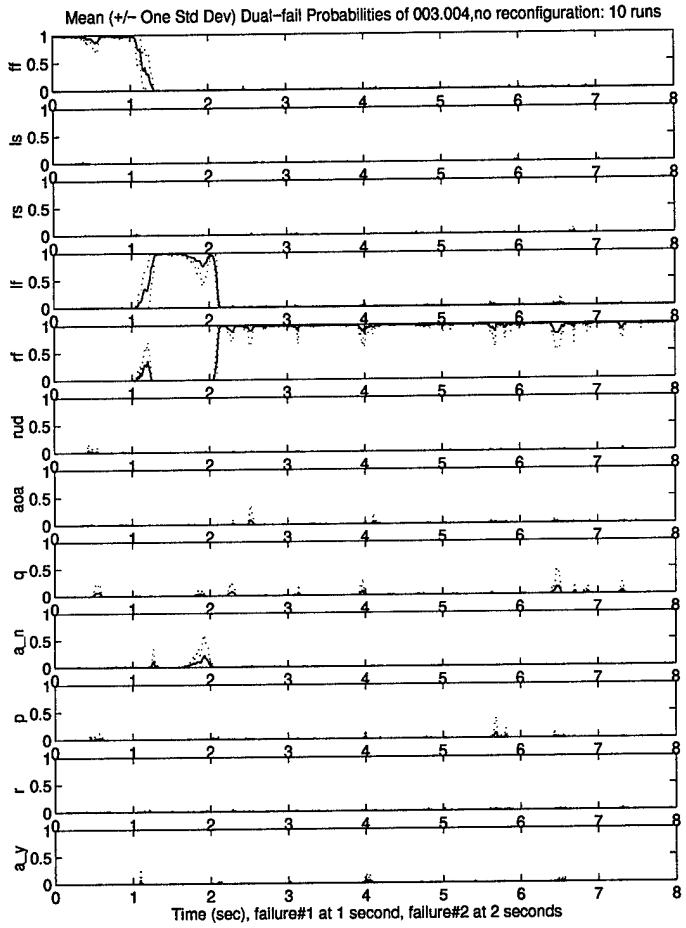


Figure 4-10 Left Flaperon/Right Flaperon Failure, Control Reconfiguration 'OFF'

Since time did not permit a massive review of all poorly rated cases in Table 4-8, in which the contribution of Control Reconfiguration to each poor rating in Table 4-8 could be defined precisely, it was decided to proceed, noting for the record that there is a problem here with MMAE detection of the second

total impairment that is caused by Control Reconfiguration and which must be analyzed in the immediate future. Though the *dual actuator impairment cases* (all cases of left stabilator as first impairments) which *were* run with the revised and expanded MMAE/CR algorithm, equaled or even outperformed Lewis' data, Lewis [26] got much better results with his implementation of "scaled Control Reconfiguration" for these cases. (This leads to a very strong recommendation made in Chapter 5 to implement CR using complete pseudoinverse calculations to avoid differences that are emerging between researchers in their respective implementations of "abbreviated" Control Redistribution calculations.) Cases of the right flaperon as the first actuator impairment (Table 4-8, row 'RF') yielded much better results than the previous cases of "left flaperon first" impairments. The "Fair" ratings that were given were due to some probability clipping starting at 2 seconds and lasting for the duration of the simulation, and probability dropouts seen in the last three seconds.

The cases of "rudder-first" total impairments (Table 4-8, row 'RUD') deserve some special attention, beginning with the case of the rudder as the only impairment (Table 4-8). That case is rated as "Poor" because the Probability Plot in Appendix D.1 shows that it has the same qualities that were seen for the rudder failure case (with CR) in Figure 4-9, namely rapidly deteriorating performance after 2 seconds, due to probability spillages onto other MMAE channels as well as probability dropouts. That description fits all entries, for the rudder first impairment case, rated as "Poor" in Table 4-8. Clipping, spillage onto other MMAE channels, and probability dropouts all combined to result in the poor ratings which were given. Nevertheless, even in the cases rated as "Poor", the MMAE detection trends were promising, in that persistent activity occurred primarily on the two correct channels. It came as a surprise that the two cases of left and right stabilators as *second* impairments showed vastly improved MMAE detection performance over the corresponding cases in which these stabilator impairments were the *first* impairments. Except for some probability spillage occurring during detection of the second impairment, along with clipping of the stabilator probabilities (10%) in each case, those two cases were almost rated as "Good", and were *almost* identical to Lewis' results. That would be an indication, of course, that the stabilators had some sort of adverse effect on MMAE detection capability *after* a rudder impairment, though a reason for that effect has yet to be found.

It is obvious that a more prudent course of research if given sufficient additional time would be to try to resolve the differences, occurring in some cases, between these and Lewis' results [26]. It would also be very wise to find the cause of the adverse effect that Control Reconfiguration has upon cases in which the rudder is the only impairment or the first of two impairments. That must be left, however, to future researchers, but some suggestions will be made in Chapter 5 ("Results and Conclusions") to aid in the search. For this thesis, time constraints dictated moving ahead with the next two stages of research.

4.12. MMAE/CR Performance During Single and Dual, 75% Actuator Impairments

This section begins by examining MMAE performance during *single*, 75% actuator impairments. Performance will be evaluated first for impairment cases with both reconfiguration and filter bank swapping *inactive*, to be followed by the same partial impairment cases with Control Reconfiguration activated (but without bank swapping). This progression will serve to isolate the effects of adding Control Reconfiguration only. Once that examination has been made, the results of dual, 75% actuator impairments and dual impairments of "75% actuator/100% sensor", *with filter bank swapping and Control Reconfiguration activated*, will be presented and compared to the single impairment cases using a tabular format with quality ratings and observations following the example of Table 4-8.

4.12.1. Single, 75% Impairments (25% effectiveness) of Actuators with Reconfiguration and Filter Bank Swapping Disabled

Figure 4-11 shows the Probability Summary Plot resulting from 10 Monte Carlo simulations of single 75% actuator impairments occurring at 1 second into the simulation, with Control Reconfiguration and MMAE filter bank swapping *disabled*. For all actuators except the rudder, it appears that MMAE may

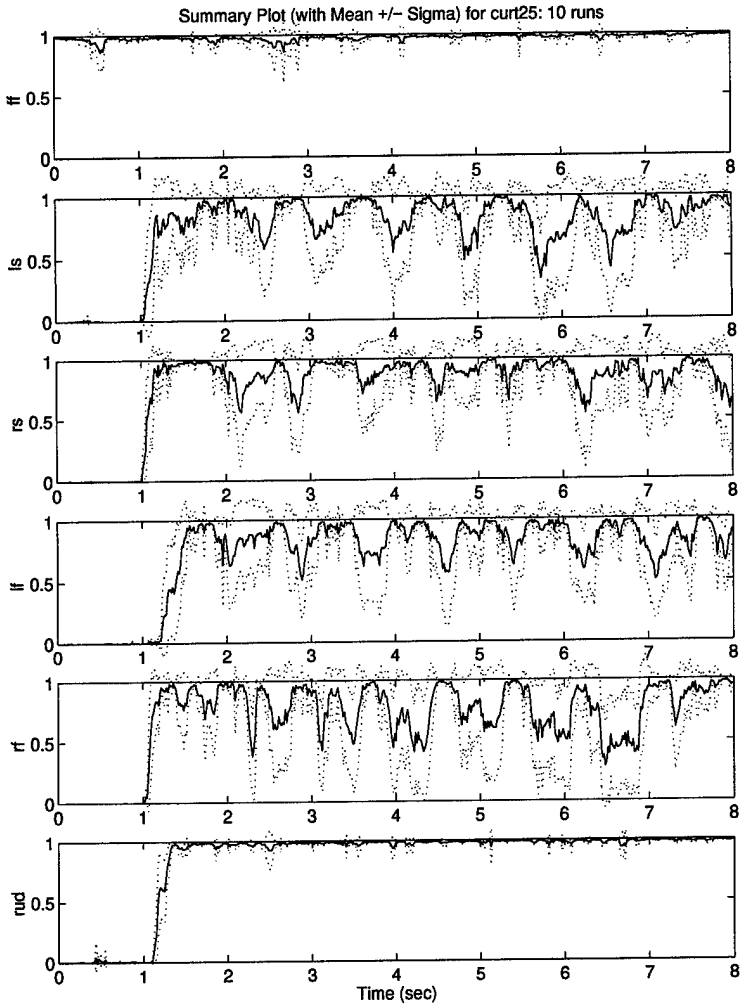


Figure 4-11 Probability Summary Plot for Single 75%Actuator Impairments : Control Reconfiguration and Bank Swapping Disabled

be doing a fair job of blending, in that overall mean values of probability from 1 to 8 seconds appear to be in the neighborhood of 0.75. While each subplot in that figure may appear, at first glance, to show fairly good performance from MMAE, it is important to view the corresponding Probability Plots in Appendix B.3 to get an accurate assessment of blending performance between the correct channels, probability drop-out, and spillage or false alarms onto other MMAE channels. To help provide that assessment, Table 4-10 provides a comparison between temporally-averaged mean values, starting at 1 second and ending at 8 seconds, of each subplot in Figure 4-11 and temporally-averaged mean probability values of all other channels on the corresponding Probability Plots. Each entry in the left-most column of Table 4-10

represents the single actuator impairment and contains the temporally-averaged mean probability value of the probability plot from that MMAE channel as seen in Figure 4-11. The 12 entries in the top row of Table 4-10 represent the 12 channels within MMAE. The number in each table entry is the temporally-averaged mean probability value of the MMAE channel from t = 1 second, when the single impairment is inserted, to the end of the simulation run at 8 seconds. What one would *wish* to see in cases of 75% actuator impairments, is a number close to 0.75, corresponding to 75% partial impairments, for each actuator listed at the left, together with a number close to 0.25 in the column labeled "FF" for "fully functional". This would mean that "perfect" blending had occurred within MMAE. We would then like to see, when looking across the columns for all other channels, values close to zero which would mean that very little if any probability activity (spillage, false alarms, etc.) had occurred.

MMAE Channels

	FF	LS	RS	LF	RF	RUD	AOA	Q	A_n	P	R	A_y
LS 0.824	0.069		0.009	0.003	0.003	0.021	0.030	0.010	0.006	0.011	0.012	0.001
RS 0.874	0.039	0.002		0.015	0.006	0.009	0.011	0.005	0.011	0.008	0.020	0.002
LF 0.810	0.097	0.002	0.002		0.010	0.004	0.008	0.004	0.034	0.021	0.007	0.001
RF 0.774	0.141	0.004	0.002	0.009		0.004	0.016	0.010	0.007	0.020	0.011	0.001
RUD 0.948	0..034	0.001	0.001	0.002	0.002		0.002	0.002	0.002	0.003	0.002	0.001

Table 4-10 Mean Probability Values of MMAE Channels: Single 75% Actuator Impairments

Control Reconfiguration and Bank Swapping Disabled

One can see from examining Table 4-10 as well as from examining the Probability Plots in Appendix B.1, that MMAE *overestimates* the amount of control effectiveness lost in all cases, especially for the rudder, which, as we already know, will require special attention and techniques for accurate estimation. We also see from the values in each row that there is some spillage onto other MMAE channels for *all* cases, but there is also a promising trend in that most of the probability *is* shared (as it should be) between the fully functional (FF) column and the (left-most) impaired actuator channel. The question we wish to ask ourselves, especially when viewing the Probability Summary (Figure 4-11) or Probability Plots, is: "given a probability plot showing an actuator impairment, could an accurate estimate be extracted from that data within the first 0.5 second after the initial detection?" That is the length of time (a figure of merit currently

accepted as a "benchmark", based upon the unstable, open loop airframe characteristics of the F-16) within which the estimate should have been extracted from the data, used to make the correct filter bank switch, and sent to the Control Reconfiguration module. When this question is considered, it appears that there may be a problem with getting an accurate estimate for most of the impairments. Except for the rudder, the Probability traces seen on the Summary Plot show fairly good blending behavior over a long time interval, but the *leading edges* of the plots for all actuators except the left stabilator, meaning the first (approximately) half second of data, show indications of *total impairment.* The left stabilator channel, on the other hand, shows the sort of behavior we would like to see for all actuators in that the probability rises rapidly to a level which looks very close to the actual level of .75 and remains there for about 0.5 seconds. This would allow a timely and accurate estimate by the quantizer and would ensure proper bank switching and reconfiguration. Although it would be a simple matter to adjust, or bias, the quantizer levels to achieve a good estimate for these partial impairment cases, such an ad-hoc method would potentially ruin the good performance seen during total impairments. One might suspect that this difficult situation, together with the behavior of the rudder probability, may make additional techniques necessary for accurate estimation of partial impairments of some or all actuators. We next examine the effects of Control Reconfiguration on MMAE blending performance, leaving MMAE bank swapping disabled.

4.12.2. Single 75% Impairments of Actuators with Reconfiguration (Filter Bank Swapping Disabled)

Figure 4-12 shows the Probability Summary Plots for the same impairment cases with Control Reconfiguration. It is easy to determine the effects due to the addition of Control Reconfiguration as seen in Figure 4-12. The leading edges and first two seconds of all probability plots for all actuators are virtually identical to the corresponding plots of Figure 4-11 until just after two seconds, when the probabilities begin much larger, strongly periodic excursions about the temporally-averaged mean values listed in Table 4-11. The amount of probability spillage onto other channels within MMAE as seen in Table 4-11 is not noticeably different from Table 4-10, except that when viewing the corresponding Probability Plots in Appendix B.2, it is apparent that Control Reconfiguration causes many more probability "spikes" onto other

MMAE channels, and onto sensor channels in particular. The rudder (rud) probability behavior seen in Figure 4-12 may lead one to believe that good blending is occurring, but upon examining the Probability Plot for that case in Appendix B.2, it is apparent that the probability is not given up to the "fully functional"

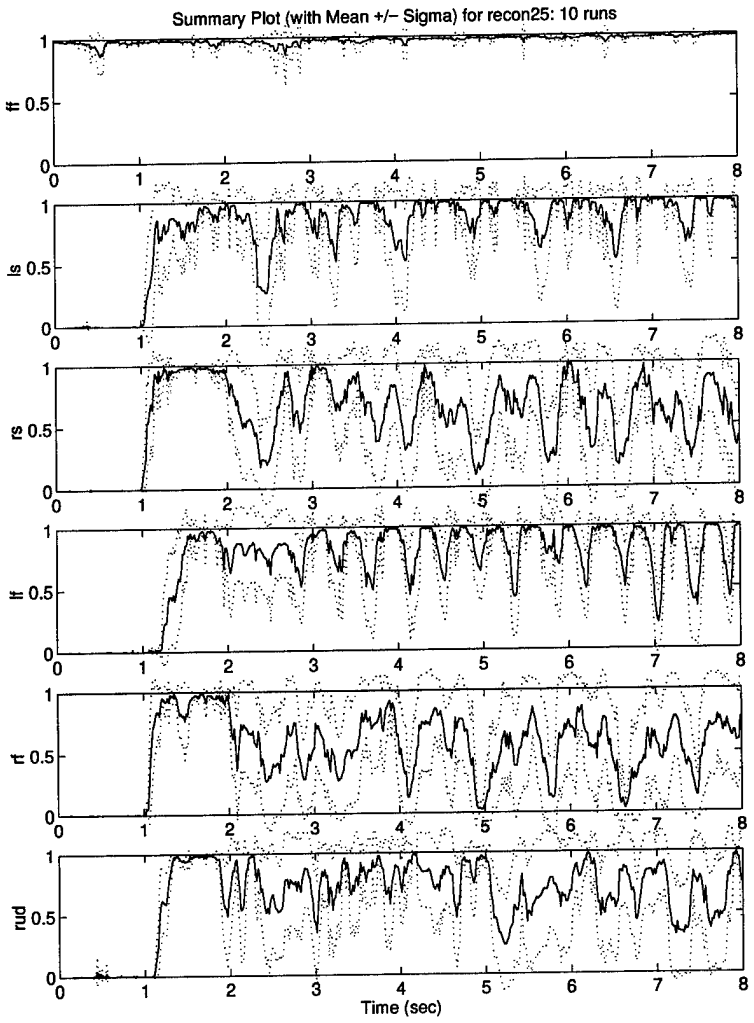


Figure 4-12 Probability Summary Plot for Single 75% Actuator Impairments With Control Reconfiguration (Bank Swapping Disabled)

(ff) channel as desired, but is lost in spillages to the yaw rate, pitch rate, and normal acceleration sensors, as well as to the left and right stabilators. The most likely cause is that the carefully-phased dither signals are becoming "scrambled" in the sense that the relative amplitudes and/or relative phases are being spoiled by Control Reconfiguration, hence causing temporary probability spillage onto sensor channels that MMAE perceives as temporary "sensor failures". It *is* worth noting here that the probability quantization scheme, with its thresholds and leading edge / trailing edge timers, will filter out most of the short duration

probability drop-outs and spikes seen on the Probability Plots, and will ignore the sensor false alarms completely since the scheme, as currently implemented, only operates on MMAE actuator channels.

	MMAE Channels											
	FF	LS	RS	LF	RF	RUD	AOA	Q	A_n	P	R	A_y
LS 0.859	0.061		0.002	0.008	0.004	0.014	0.027	0.004	0.002	0.007	0.010	0.001
RS 0.668	0.118	0.002		0.022	0.011	0.027	0.052	0.005	0.010	0.013	0.079	0.002
LF 0.789	0.122	0.002	0.002		0.012	0.015	0.017	0.015	0.014	0.011	0.010	0.001
RF 0.584	0.301	0.003	0.003	0.003		0.004	0.028	0.005	0.034	0.016	0.017	0.002
RUD 0.005	0.735	0.042	0.010	0.009	0.004		0.018	0.062	0.017	0.005	0.088	0.004

Table 4-11 Mean Probability Values of MMAE Channels: Single 75% Actuator Impairments; With Control Reconfiguration.

4.12.3. A Pause to Reflect

A question which should be asked at this point is: "Why proceed to the dual impairment cases when there are known problems with properly detecting and estimating single partial impairments?" The issue is once again, one of time. If sufficient, additional time were available to explore various methods of compensation, it *would* be wise at this point to correct or compensate for known problems with MMAE performance in detecting and estimating single, partial actuator impairments. We know by now that blended probabilities from the MMAE algorithm are biased with respect to actual impairment values. Furthermore, we can expect that the probability quantization scheme will pass these biased estimates on to the decision logic used for bank swapping as well as to the Control Reconfiguration module, unless adjustments and compensation techniques are employed. The effect will be a transition to an incorrect "Level 1" MMAE filter bank as well as an incorrect "reconfiguration solution" after the first partial impairment, and this will no doubt have consequences on detection of second actuator impairments. Having acknowledged the problems inherent in the existing algorithm, it was decided to proceed to dual impairments and to uncover all possible insight, problems, and issues for future research to address.

4.12.4. *Dual 75% Impairments of Actuators, Bank Swapping and Control Reconfiguration Activated*

Results of dual, 75% actuator impairments and of "75% actuator/100% sensor" impairments are summarized in Table 4-12, from interpretation of the associated Probability Plots in Appendix D.2. (Probability Summary Plots, of the type presented for cases of single actuator impairments, are not presented since they have little meaning in these cases of dual partial impairments. Compilation of such a plot for each row of Table 4-12, requires the selection of one dual impairment case to represent the probability profile of the first impairment, and that would be misrepresenting the facts since the probability trace of the first impairment *differs* between each second impairment case.) We look first at the diagonal **BOLDFACE** entries in Table 4-12 which give an indication of MMAE filter bank swapping performance when there is only the first partial 75% actuator impairment, and when bank swapping is activated. All entries are rated "Poor", meaning that, after the first impairment detection, estimation, and bank swap to *some* "Level 1" filter bank, there were numerous false alarms and losses of probability from the impaired actuator to other channels. What should have been observed was a probability "trade-off" only between the "fully functional" channel and the channel belonging to the "actually impaired" actuator. We strongly suspect that what has occurred has been an erroneous transition to the *incorrect* "Level 1" filter bank, most likely to a bank corresponding to the first impairment being declared as totally impaired. (We cannot say

Second Impairment

	LS 75%	RS 75%	LF 75%	RF 75%	RUD 75%	AOA 100%	Q 100%	A_n 100%	P 100%	R 100%	A_y 100%
LS 75%	**POOR**	Poor*	Fair*	Fair*	Fair*	Good	Good	Fair	Fair	Fair	Good
RS 75%	Poor*	**POOR**	Poor*	Poor*	Poor*	Poor*	Fair*	Poor*	Poor*	Poor	Good
LF 75%	Poor	Poor	**POOR**	Poor*	Poor	Poor	Poor*	Poor*	Poor*	Poor	Poor*/ Good
RF 75%	Poor	Poor	Poor	**POOR**	Poor	Poor	Poor*/ Good	Poor*/ Good	Poor*/ Good	Fair	Fair*/ Good
RUD 75%	Poor*	Poor*	ND	ND	**POOR**	Poor*	Poor*	Poor*	Poor*	Poor*	Poor*

Table 4-12 Dual Impairment Evaluation: 75% Actuator/75% Actuator Impairments

75% Actuator/100% Sensor Impairments

Rating	Qualitative Description of Ratings for Dual Impairments: 75% Actuator/75% Actuator 75% Actuator/100% Sensor
Good	Second detection is on correct MMAE channel and shows performance equal to or better than first. Good probability lock and hold after second impairment. Virtually no probability "spillage" onto other MMAE channels. Good estimate for Control Redistribution
Fair	Second impairment displays performance degraded from single impairment case. Temporary but significant dropouts from probability lock. Some "spillage" onto other MMAE channels. Could get fair estimate for Control Reconfiguration.
Poor *	Performance drastically degraded from single failure case because of frequent spikes and drops in probability with no lock. Accurate estimate for Control Reconfiguration unlikely. Also marked by excessive "spillage" onto other channels or "indecision" by MMAE on a sample by sample basis Asterisk (*) means that trends/probability profiles were very promising in spite of rating given
ND	No detection, or an incorrect MMAE detection for second impairment
BOLDFACE	Rates MMAE ability to continue detecting and estimating first partial impairment when no second impairment is inserted, but filter bank swapping has occurred.

Table 4-13 Rating Scale for Table 4-12

this with absolute assurance because the banks actually swapped may vary slightly in a 10 Monte Carlo run simulation.) This guarantees that there will be errors in all filters within the bank and that false alarms will ensue, as is evident when viewing the Probability Plots. This has also most likely caused an improper control reconfiguration solution based on a 100% actuator first impairment, which compounds the first problem even further. It would be instructive, for future research, to repeat these cases, artificially *forcing* proper bank swapping to occur after the first impairment and to view the improvement in results. Not only would this be a valuable performance benchmark against which to compare actual bank swapping performance while troubleshooting, but it would also serve to verify the design of the associated "Level 1" Kalman filters.

Examining the dual impairment cases in Table 4-12, it seen that many entries rated as "Poor" are followed by asterisks (*). This means that, in spite of the probability drop-outs, spillages, and false alarms onto other, "incorrect" channels, there was a definite probability "profile" between the proper MMAE channels which approximated expected behavior, and would lead one further to suspect that better MMAE performance was attainable. Taking the case of "(LS,RS)" from Table 4-12, meaning a *left stabilator 75% impairment* at 1 second followed by a *right stabilator 75% impairment* at 2 seconds, we see a rating of "Poor*". The Probability Plot in Appendix D.2 for this example shows a good, first detection at 1 second. We can assume that by 1.5 seconds, bank swapping has occurred and Control Reconfiguration has taken place. We suspect improper bank swapping might have occurred, since, from the time of the second

impairment at 2 seconds until the simulation ends at 8 seconds, there is probability spillage and large spikes on other MMAE actuator and sensor channels, even though there is more persistent probability activity on the *correct* channel. The numerous cases in Table 4-12 followed by asterisks lead one to conclude that the *maximum potential of the existing MMAE algorithm* in detecting dual partial impairments is far from being realized. Table 4-12 also contains some cases with two ratings separated by a slash, the better rating (underneath) applying if the "two-second argument", presented in Section 1, applies. It is left to the reader to examine additional cases (Appendix D.2) of specific interest listed in Table 4-12. The logical conclusion that should be apparent from these results is that everything depends on the correct estimation of the first impairment, and the "swap" to the correct filter bank. This *may* be accomplished by extensive readjustment, or biasing, of the threshold values in the probability quantizer. "Calibration" of the existing algorithm to *estimate the first partial impairment accurately, without spoiling total failure performance*, could easily become a complete research topic by itself. Regardless of how much readjustment or calibration is attempted, it is felt that the real problem to overcome will still be the tendency of MMAE blended probabilities, *in the first half second* after detection, to show "totally failed" behavior. It is unfortunate that, within that same half-second time period, it is currently accepted as "essential" to extract the impairment estimate and to reconfigure the actuator commands. (This is debatable. A real requirement should be based upon the open-loop-aircraft, "time to double amplitude", which is easily obtained from the eigenvalues of the linearized plant.) As stated before, this may require that other measures be taken such as the design of filter sub-banks, between MMAE filter Levels 0 and 1, to resolve the detected impairment further before a bank swap is made to the dual-failure filter banks. In other words, perhaps *"the cure for the ills of MMAE is more MMAE."*

4.13. MMAE/CR Performance Results for Single and Dual, 50% Actuator Impairments and 50% Actuator/100% Sensor Impairments

This last group of research results, follows the format of Section 4.12 to present results of MMAE/CR algorithm performance during dual, 50% impairments of actuators and "50% actuator / 100% sensor" dual impairments. The results of single, 50% actuator impairments *without Control Reconfiguration*

or filter bank swapping are presented in Section 4.13.1 first, followed by the same cases with Control Reconfiguration activated in Section 4.13.2. Section 4.13.3 then presents the 50% dual impairment cases.

4.13.1. Single 50% Impairments of Actuators with CR and Bank Swapping Disabled

Figure 4-13 shows the Probability Summary Plot for single, 50% actuator impairments inserted at 1 second, with *no Control Reconfiguration* and with *MMAE filter bank swapping disabled*, while Table 4-14 summarizes the temporally-averaged mean probability values of the 12 MMAE channels beginning at 1 second, when the impairment is inserted, and ending at 8 seconds. Again, this table is meant to complement the Probability Plots for these cases (Appendix C.1), and is a broad numerical summary of probability trade-off between the two appropriate channels and of probability "spillage" occurring onto other MMAE channels.

These results show a very noticeable deterioration in MMAE blending performance from the 75% impairment cases presented in Section 4.12.1. The probabilities for all surfaces except the rudder show strong periodic behavior, equal to the dither frequency, about a very low temporally averaged (1 to 8 seconds) mean value as tabulated in Table 4-14. While the left and right stabilator channels show a fair amount of blending when viewed over the entire simulation time period, it is apparent that special measures will have to be taken to process the probability data and extract the quick (within 0.5 seconds) and accurate estimate required to switch to the correct MMAE "Level 1" filter bank and to reconfigure. Due to time constraints, that is beyond the scope of this thesis research. Hence, it is a virtual certainty that these impairments will not be processed by the probability quantizer and bank swapping logic, and that a successful swap to the proper "Level 1" bank will not occur. On a more promising note, Table 4-14, rows 1 and 2, columns 1 and 2, show that the required blending *is* taking place within MMAE *between the fully functional and stabilator channels*, with relatively little spillage onto other MMAE channels. MMAE

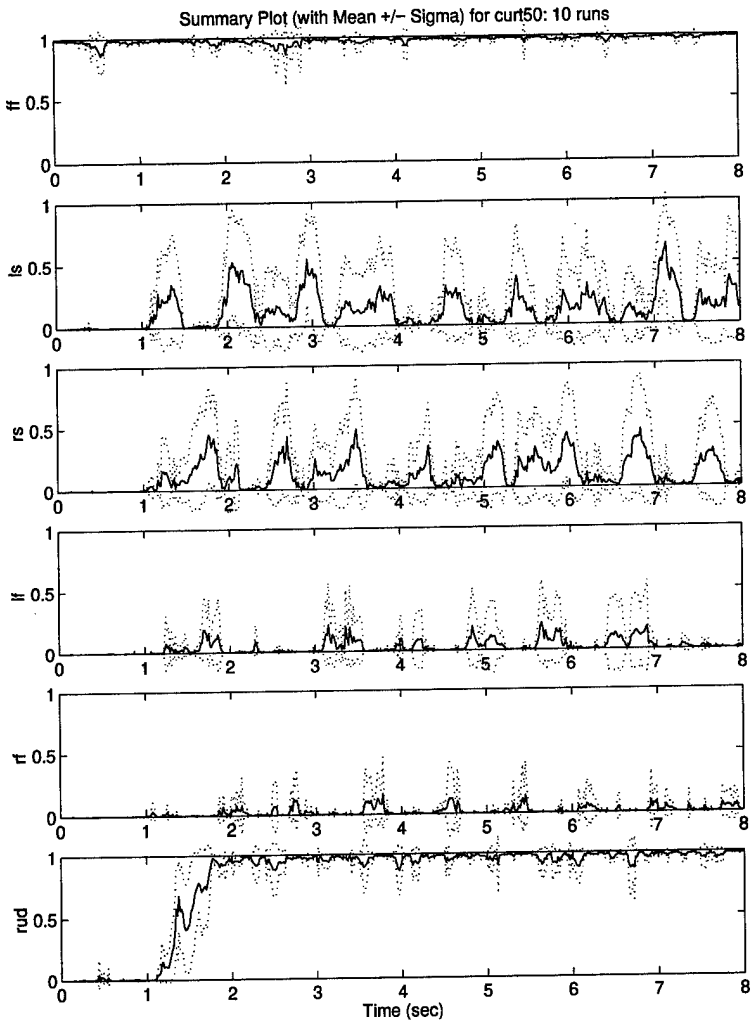


Figure 4-13 Probability Summary Plot for Single 50%Actuator Impairments; Control Reconfiguration and Bank Swapping Disabled

MMAE Channels

	FF	LS	RS	LF	RF	RUD	AOA	Q	A_n	P	R	A_y
LS 0.141	0.773		0.009	0.002	0.002	0.014	0.022	0.010	0.004	0.010	0.011	0.001
RS 0.122	0.780	0.002		0.003	0.002	0.008	0.005	0.009	0.014	0.009	0.045	0.001
LF 0.036	0.856	0.001	0.002		0.003	0.002	0.004	0.004	0.080	0.007	0.005	0.001
RF 0.018	0.945	0.002	0.001	0.002		0.001	0.002	0.003	0.015	0.005	0.004	0.001
RUD 0.900	0.071	0.002	0.002	0.002	0.002		0.005	0.004	0.003	0.005	0.003	0.001

Table 4-14 Mean Probability Values of MMAE Channels: Single 50% Actuator Impairments, Control Reconfiguration and Bank Swapping Disabled

detection ability given left and right flaperon impairments at 1 second has dropped off dramatically from the 75% impairment cases (Section 4.12.1). Probability traces are now almost "in the noise" level, evidenced by the values in Table 4-14, rows 3 and 4, columns 1 and 2. It is a guarantee that, like the stabilator impairments, MMAE filter bank swapping will not occur for these impairments. The rudder is still declared decisively as "fully failed" by MMAE, although by now the leading edge of the probability plot shows that it happens at a slower rate. It has been said already that special techniques will have to be examined for partial rudder impairment detection and estimation, but we can expect that, given these characteristics, it is likely that MMAE will switch to the incorrect "level 1" filter bank when bank swapping is activated. Some overall, interesting trends to note from Figure 4-13 are that the probability traces for the left and right stabilators are approximately equal but 180 degrees out of phase with each other. The same can be said for the flaperons, but it is noticed that they are "opposing" the stabilators, in the sense that the left flaperon probability trace is in phase with the right stabilator, and the right flaperon is in phase with the left stabilator. To someone trying to find the exact cause of this periodic behavior seen in the probability plots, these characteristics suggest an extensive examination of the separate effects of each dither signal, and whether it is feasible to schedule different sequences of dither signals based on the *initial* detection by MMAE.

4.13.2. Single 50% Impairments of Actuators with Control Reconfiguration (Bank Swapping Disabled)

Figure 4-14 shows the Probability Summary Plot for single 50% actuator impairments inserted at 1 second when Control Reconfiguration is activated. MMAE filter bank swapping is still inactive, so that the effects on MMAE performance due solely to the addition of Control Reconfiguration can be evaluated.

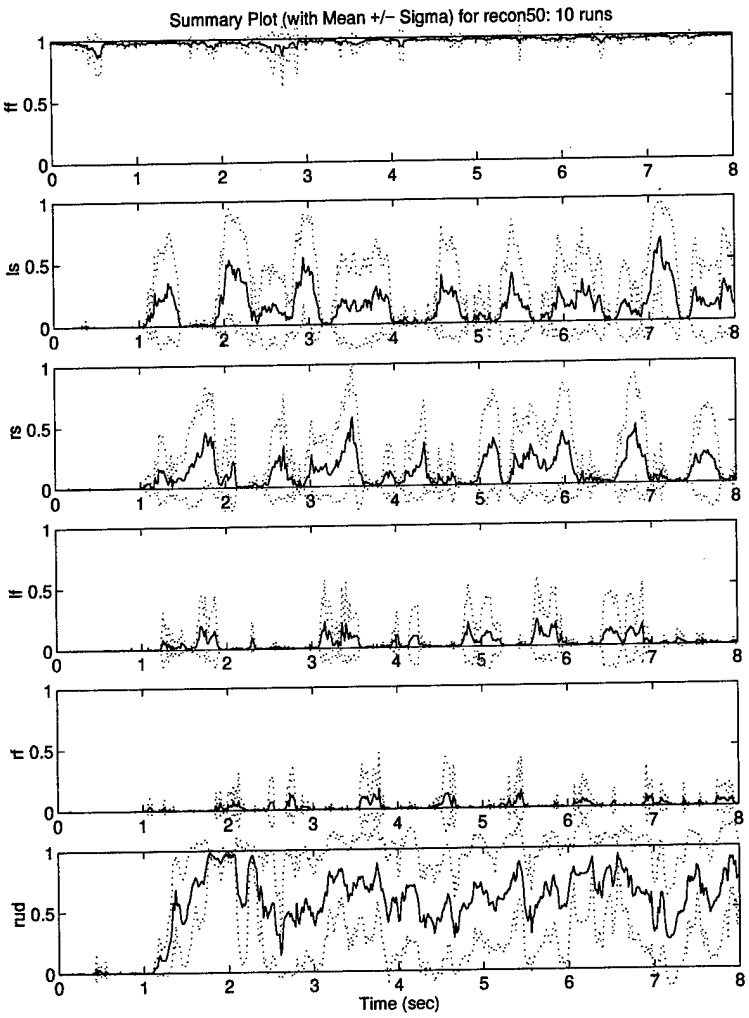


Figure 4-14 Probability Summary Plot for Single 50% Actuator Impairments With Control Reconfiguration (Bank Swapping Disabled)

These results confirm the predictions, made earlier, that the probability quantization module within MMAE will miss the detection of all 50% impairments, except the rudder. Due to extremely low amplitude probability activity below the minimum quantization level of 0.5, the quantized probability estimate upon which Control Reconfiguration is based, is zero. (With ad-hoc biasing, this *might* be compensated.) Figure 4-14 shows that the probability traces for stabilators and flaperons with reconfiguration are fundamentally the same as the cases without reconfiguration. Table 4-15 verifies this further and shows that

MMAE Channels

	FF	LS	RS	LF	RF	RUD	AOA	Q	A_n	P	R	A_y
LS 0.141	0.767		0.010	0.002	0.002	0.016	0.020	0.008	0.004	0.009	0.019	0.002
RS 0.177	0.779	0.002		0.003	0.002	0.008	0.006	0.009	0.013	0.010	0.045	0.001
LF 0.036	0.858	0.001	0.002		0.003	0.002	0.004	0.004	0.079	0.007	0.005	0.001
RF 0.018	0.945	0.002	0.001	0.002		0.001	0.002	0.003	0.015	0.005	0.004	0.001
RUD 0.777	0.081	0.014	0.015	0.006	0.006		0.012	0.024	0.011	0.005	0.047	0.004

Table 4-15 Mean Probability Values of MMAE Channels: Single 50% Actuator Impairments with Control Reconfiguration (Bank Swapping Disabled)

the numerical comparisons for all actuators except the rudder are almost identical to cases without reconfiguration. Since the probability quantization module of the MMAE algorithm is not being triggered, then Control Reconfiguration is not occurring. An examination of the state plots in Appendices G.1 and G.2 verifies this hypothesis, showing that control surface positions are identical between cases with and without Control Reconfiguration. The MMAE rudder channel, on the other hand, shows that probability quantization and Control Reconfiguration *is* occurring for an actual 50% rudder impairment. However, Control Reconfiguration *may* be occurring based on the *correct* quantized estimate or, a more likely case, upon some other estimate. We are unsure of the *actual* estimate since we don't know the exact quantized probability value for every Monte Carlo run (though that information *could* be extracted from existing data with sufficient time). This is most likely causing the carefully crafted control dither signal to become "scrambled", and hence causing the massive probability dropouts and spillage seen in the Probability Plot for this case (Appendix C.2). On a positive note, it is apparent that we have now picked up sensitivity in the rudder (rud) channel to the true level of partial impairment, since MMAE no longer indicates full rudder failure. Some valid suggestions for successful detection and reconfiguration in cases of 50% actuator impairments may be to "boost" or bias the raw probability values from the MMAE channels to values which may then be quantized to yield an impairment estimate for Control Reconfiguration purposes. Or, to get away from ad-hoc approaches to a method which may bear more fruit, one may set "trigger thresholds" for each MMAE probability channel. Exceeding those thresholds would "wake up" a "filter sub-bank" within

MMAE with the express purpose of "resolving" the single, partial impairment to an accurate discrete value upon which to swap banks and reconfigure.

4.13.3. Dual, 50% Impairments of Actuators and Dual Impairments of "50% Actuator and 100% Sensor Combinations" with Bank Swapping and Control Reconfiguration

These dual impairment cases yielded results very different from the total impairment cases presented in Section 4.11.3 and the 75% impairment cases in Section 4.12.4. In many dual impairment cases, except for cases of 50% rudder actuator first impairments, the probability quantizer and filter bank swapping logic missed the first impairment occurring at 1 second but detected the second impairment at 2 seconds. Thus, "from the MMAE's point of view" (the Probability Plots), many impairments occurred in reverse, especially in cases where the second failure was a 100% sensor failure. This was not counted "as a defeat for MMAE", since success is measured primarily by detecting the *correct* impairments, and specific *order* is of secondary importance. It was also not a *total* surprise that the impairments were detected in reverse: full sensor failure always has been quicker and easier to detect than a partial actuator failure. The tabular formats used in Sections 1 and 2 to rate the quality of MMAE performance during dual impairments are no longer useful for comparing this data to earlier results, since, strictly speaking, every case would be rated an utter failure. Instead, the results are presented in groups based on the first 50% actuator impairment.

4.13.4. 50% Left Stabilator (LS) First Impairments

For the case of no second impairment, the Probability Plot (Appendix D.3) shows that MMAE is performing as if bank swapping were disabled. The left stabilator (ls) probability trace is identical to the trace for the left stabilator in Figure 4-14, meaning that no bank swapping is occurring in MMAE, that quantized probabilities, upon which bank swapping depends, must be zero, and that no Control

Reconfiguration is occurring. We can expect at this point that all of the following cases of "left stabilator first" impairments will miss the first (left stabilator) detection completely, unless additional, compensating techniques are employed to detect the first impairment. The probability plot shows that a good probability tradeoff exists between the fully functional and left stabilator channels from 1 second, when the impairment is inserted, until about 3 seconds, when probability begins to spill onto the rudder channel and to a lesser extent onto all sensor channels. The spillage is not a great concern, however, when rationalizing that detection, estimation, and reconfiguration should have occurred within fractions of seconds after the initial detection. Some possible answers to detecting and estimating this impairment properly are to add additional, lower thresholds to the probability quantizer and then "upwardly bias" the bank switching algorithm, i.e. if the quantizer detects a level of .25, for example, then switch to the bank for an impairment of .5. Another technique would be set a very low threshold value which the raw probability value coming from MMAE would "trip" to obtain a "wake up" signal, and using that signal to activate a "partial impairment resolver", as advanced and described in earlier sections.

For the remaining cases of 50% left stabilator first impairment followed by a second, 50% impairment of any other actuator, it is apparent by examining the "fully functional" (ff) channel of the probability plots that the second impairment is detected first. Perhaps it is more appropriate to say that the second impairment, *in addition to the first impairment*, causes the fully functional channel of MMAE to drop to a much lower probability value. MMAE still cannot distinguish between the impairments decisively, as evidenced by the large amount of probability spillage onto other channels, although there is a hint of greater MMAE performance potential. If one confines attention to only three of the twelve subplots on any of the probability plots for these impairment cases, namely the fully functional (ff) subplot and the two channels for the two impaired actuators, then it is seen in the probability traces that there is a definite indication, or recognition by MMAE, of the two impairment "events".

There were six cases of a 50% first impairment of the left stabilator actuator, followed by a 100% sensor impairment. These were very interesting cases, because in all of them, the sensor impairment was detected strongly enough by MMAE to trigger a bank swap to a "Level 1" filter bank hypothesizing the sensor failure. Since there were not any partial failure filter banks for sensors (because partial sensor failures were not modeled in this thesis), it was guaranteed that a bank swap would be made by the MMAE

algorithm to a filter bank well-modeled for one of the actual impairments. After the apparent bank swap, MMAE continued to estimate the partial actuator impairment *but with strikingly better performance* than that seen just previously when the second impairment was an actuator. The probability plots for these cases show virtually all probability activity now confined to the proper channels for the actual impairments, and serve to illustrate the performance which may be forthcoming when the problems with improper bank swapping encountered in Sections 4.11.3 and 4.12.4 are finally solved.

4.13.5. 50% Right Stabilator (RS) First Impairments

This group of results was virtually a mirror image of the first group of "50% left stabilator first" impairments. Detection of the actual first impairment was missed by the probability quantization and bank switching logic entirely, and for cases of a second 50% actuator impairment, probability plots showed indecision and false alarms by MMAE, though the same promising trends were noticed that were noted in the previous subsection. When the second impairment was a 100% sensor impairment, the probability plots showed much better, "cleaner" performance by MMAE, which continued to estimate the original 50% actuator impairment. It should also be noted at this point that, although the probability traces were of much better quality, they were still not thought to be good enough for extraction of an impairment estimate directly, but could certainly be used to trigger other, enhanced estimation schemes.

4.13.6. 50% Left Flaperon (LF) First Impairments and 50% Right Flaperon (RF) First Impairments

These two groups of results closely followed the trends and examples set by the first two groups of 50% stabilator first impairments, but with two very interesting cases worth noting for further analysis. In the dual impairment cases, "50% left flaperon / 50% rudder" and "50% right flaperon / 50% rudder", it was observed that except for some barely noticeable probability activity, neither impairment was detected. This was surprising given the earlier tendency of MMAE, during single impairment tests, to estimate the rudder

as "fully failed" for actual impairment values of 50%. It was expected that the 50% rudder actuator impairment would appear, and there is no known reason for that not to occur here, unless there is some insight missing as to the effect the flaperons have upon an accurate rudder impairment estimation. Again, if sufficient additional time were available, this would be a relationship worth establishing. For instance, perhaps decreasing the amplitude of the roll dither signal would allow a better estimate of rudder impairments.

4.13.7. 50% Rudder (RUD) First Impairments

This group of dual impairment tests yielded results a bit different from those seen in the first four groups. Because the current MMAE algorithm has a strong tendency to estimate the rudder as "fully impaired" for actual impairments of 100% to 50%, the first impairment of the rudder actuator *was* detected by the probability quantization and bank switching algorithm, and Control Reconfiguration *did* occur. The problem with this group is seen in the resulting Probability Plots in Appendix D.3. All of those plots show the rudder actuator impairment occurring at 1 second, followed by an increase to an incorrect probability value of 1.0 on the rudder (rud) channel. However, after two seconds, one sees a change in the appearance of the rudder probability subplot for all cases, including the case of no second impairment. Specifically, one observes rudder probability peaks being "clipped" at apparently the same level (and this level appears to be discretized to an integer number of tenths). This feature must be related in some way to having activated filter bank swapping, since clipping is not present in the plot of single 50% rudder impairment with Control Reconfiguration in Figure 4-14. What the clipping most likely means is that some detection other than a rudder detection is being made *between individual runs* within a 10 Monte Carlo run simulation, rather than just "spilling" between MMAE channels within the *same single run*, which would yield the curvilinear appearance we are used to seeing in probability plots. What is bothersome in the probability plot for the single rudder impairment is that the "missing probability" appears to be dispersed among all other channels, including the fully functional channel, so an accurate assessment of this "other detection" causing such clipping is impossible.

This feature having been noted, an examination of Probability Plots for all cases except for flaperon second impairments reveals that, once again, some promising trends are present, in that persistent probability activity is occurring primarily on the correct MMAE channels. While it is true that there is spillage that, in many cases, instantaneously equals or exceeds the probability value of the second impairment, such spillage is either random or periodic with the dither frequency, and is not persistent. In cases of 50% flaperon second impairments, the flaperon probabilities are not of any distinguishable value, but it is noticed that these impairment cases appear to take away much of the clipping seen on the rudder channel in all other cases. Though exact causes for this novel behavior have not yet been found, this group of impairment cases inspires curiosity and is a good candidate for repeating some or all cases, ensuring a perfect bank swap and correct reconfiguration by artificially providing the algorithm with perfect information for those operations. Though it is not known what cause and effect relationship there might be between "clipping" and incorrect bank swapping, the two are related and it would be helpful to assess the results.

4.14. Chapter Summary

This chapter began by presenting results of initial tests to explore the ability of MMAE to blend probabilities and produce an accurate estimate of a partial actuator impairment. These results highlighted some expected and unexpected performance problems, most notably with the appearances of the probability plots as well as with temporally-biased trend behavior. This resulted in a combination of methods, discussed in this chapter, to smooth and then quantize the actuator probabilities to try to extract an estimate from the data in a timely manner. It was also necessary to choose the discrete levels of partial actuator impairments to be modeled in the new banks of MMAE "partial impairment / total impairment" filters, which were designed and implemented in this thesis. An attempt to correct a problem with partial detection of rudder impairments, by retuning the MMAE rudder channel using two methods, was described. That problem is still present and will require special attention and methods, but some inaccuracies with other actuator detections may be correctable by ad-hoc techniques. The design and implementation of the means to

transition, or "swap" from "Level 0" to "Level 1" filter banks within MMAE was presented, and it is here that some of the ad-hoc, "biasing techniques" may be applied in order to ensure that MMAE swaps to the correct filter bank. Results of single actuator impairments, dual actuator impairments, and dual combinations of "actuator / sensor" impairments were presented and evaluated next in this chapter, and though results may not have been as good as originally predicted, the reasons for them and courses of appropriate remedial action to correct them were presented and discussed.

5. *Conclusions and Recommendations*

5.1. Chapter Overview

This chapter draws appropriate conclusions and presents recommendations based on the results presented in Chapter 4. These conclusions and recommendations are presented in sections, and in the order that the corresponding processes occur within the MMAE/CR algorithm. Sections 5.2 and 5.3 present conclusions and recommendations related to MMAE blending, detection and estimation capability of single and dual actuator impairments, followed by conclusions and recommendations related to (in order): probability smoothing and quantization, MMAE filter bank swapping, and control reconfiguration in Sections 5.4 through 5.6. Additional recommendations will be made in Section 5.7 on important check-cases and modifications to be examined prior to further research, and on improving the thesis research environment itself. Suggested topics for follow-on or parallel research efforts are presented in Section 5.8.

5.2. MMAE Blending, Detection and Estimation Performance

5.2.1. Single Partial Failures

The results of Section 4.5 demonstrated that the MMAE algorithm used in this research does, in general, have the ability to blend the probability of a "fully functional aircraft" (ff) and the appropriate, fully failed probability corresponding to the actuator impairment, to yield *some* estimate of partial actuator impairment. There are problems, however, which must be compensated or overcome, before the full ability of the current MMAE algorithm to estimate partial impairments *correctly* can be realized. Since the integrity of the MMAE/CR algorithm depended so heavily upon the correctness of the first actuator impairment estimate and reconfiguration, special attention will be given to MMAE performance during the first impairment.

5.2.2. *Partial Rudder Actuator Impairment Estimates*

First, it is obvious that the ability of this MMAE algorithm to estimate partial rudder actuator impairments is wholly inadequate. For impairments ranging from 100% to 50%, the algorithm estimates the impairment as 100%, resulting in an incorrect reconfiguration and severely hindering MMAE performance thereafter. Two methods of tuning the rudder channel (Section 4.9) in an attempt to correct the problem did not appear to have the desired effects. A decision must be made, therefore, whether to accept this problem and to try compensating for it in an "ad-hoc" fashion, or to design additional "Level 0" Kalman filters hypothesizing partial impairments of the rudder actuator. One "ad-hoc" method of compensation might involve assuming a 50% rudder actuator impairment, for *any* detection of rudder impairments by MMAE, and reconfiguring. (This suggestion, made by Maybeck [37], was explored and gave good results in general, in that it did not adversely impact MMAE performance in detecting a second impairment.) Another option is to accept the 100% estimate from MMAE and zero the command to the rudder, even though there might be 50% rudder capability remaining. Both suggestions are, of course, compromises in attainable performance from the overall MMAE/CR scheme. This thesis, therefore, recommends facing the MMAE performance problem directly and to design at least 1, and up to 3, additional Kalman filters for inclusion in the MMAE "Level 0" filter bank. Building and including additional filters is easy, and initial, proof-of-concept experiments should be performed by building one additional filter hypothesizing a 50% rudder impairment. Tests should be run of actual 75% and 50% partial rudder impairments, and observations made on improvements (hopefully) in MMAE ability to estimate partial rudder impairments. If these tests show good results, then additional filters in "Level 0" for 25% and 75% partial impairments should also be considered. Another option for including additional filters is to keep the number of Kalman Filters in "Level 0" at 12, but once an estimate of 100% rudder actuator failure is detected by MMAE, switch to a "short bank" (of 3 or 4 filters instead of 12) of partial rudder impairment filters for further refinement of the estimate, and *then* switch to the appropriate "Level 1" filter bank. Both options would be easy to implement.

5.2.3. *Partial Impairment Estimates of Other Actuators*

For actuators other than the rudder, the MMAE algorithm does display an ability to yield good estimates of partial actuator impairments. Table 4-3 shows that, over the time period of from 1 second, when the impairment is inserted, to 8 seconds, when the simulation ends, the temporally averaged probability values of full actuator failure decrease as they should for decreasing percentages of actuator impairment. Despite the promising trends, there are issues needing solutions.

One issue is that we cannot wait for the length of time it takes (7 seconds) to arrive at the estimates. We need an accurate estimate of partial impairment from MMAE as quickly as possible so that Control Reconfiguration is accomplished within 0.5 seconds (a currently accepted figure of merit) after the impairment if possible, and certainly within 1 second[1]. The Probability Summary Plots for single, partial actuator impairments of 90% (effectiveness, $\varepsilon = .1$) and 75% ($\varepsilon = .25$) show that, within the first second after an impairment, many probabilities display the same behavior as fully failed cases. It takes appreciable time for enough "peaks and valleys" to occur in the probabilities so that the mean probability value is recognizable as a partial impairment. By that time, however, a total impairment has been declared, the MMAE algorithm has already switched to the incorrect "Level 1" filter bank, and Control Reconfiguration has occurred based on an incorrect estimate. This problem is probably the single most important one to solve before continuing research into dual partial impairments, since the performance of MMAE in detecting a second impairment of any kind is extremely dependent upon the accuracy of the first estimate and proper bank switch. To attempt to solve this problem, the recommendation is to leave the probability quantization algorithm (including threshold levels and leading / trailing edge time constants) as it is and to experiment with biasing the MMAE bank swapping logic, which can can be adjusted, or *effectively* "biased", so that at the end of 0.5 seconds the quantized probability value can be calibrated to the known impairment. For instance, if an *actual* 50% partial actuator impairment is consistently *declared* as a 75% impairment, then such an indication should be used to switch to an MMAE "Level 1" filter bank for a 50%

---

[1] This is usually based on the "time to double amplitude" for the unstable, open loop F-16 airframe

first actuator impairment. A 30-sample (.5 second) window average of these probabilities before biasing may further enhance this suggestion. If this does not give positive results, then another suggestion is to try what was suggested for rudder impairments. Adding more Kalman filters to "Level 0" for partial failures, or using the initial tendency to "over-declare" the impairment to trigger an intermediate "short bank" of single, partial impairment Kalman filters may yield rewarding results. It has also been suggested to restart MMAE completely from "Level 0", in the event of a suspected mis-declaration and filter bank swap, and allow it to re-declare the first impairment. Such a reset might be triggered by exceeding a threshold set for *any* substantial activity (and not just a strong, consistent, or "hard" declaration) on the "fully functional" MMAE channel. If a reset was executed by MMAE, then MMAE might "remember" to bar the current, incorrect filter bank from consideration for a future bank swap on the same actuator. This suggestion might be examined, though it would be reasonable to expect MMAE to repeat the same mistakes within a single Monte Carlo simulation run as it does over the course of 10 Monte Carlo runs.

5.3. *Dual Partial Actuator Impairments*

It is concluded in this thesis that correct detection and accurate estimation of dual, partial actuator impairments will be vastly improved once the problems encountered during single, partial impairments are corrected. This conclusion can be reached by examining Probability Plots for cases of dual, partial actuator impairments of 75% (Appendix D.2) versus the Probability Plots (in Appendix D.1) for dual, total actuator impairments. One can surmise from those plots that, in the 75% cases, the first impairment was probably not declared and estimated correctly, and that the switch to the proper "Level 1" filter bank likely did not occur. The effects were manifested in false alarms on many MMAE actuator and sensor channels. Contrast those results with dual, total actuator impairments, where it highly probable in all cases that proper estimation and filter bank switching *did* occur, and the differences in second impairment detection quality are apparent. Since the time available for this research did not permit it, a suggestion is made to explore (briefly) the existing capability of this MMAE algorithm to detect and estimate second partial actuator impairments by giving the MMAE algorithm the correct estimate of the first impairment, thereby guaranteeing a correct

bank switch to "Level 1" and correct control reconfiguration, and observing the results when 90%, 75%, and 50% second actuator impairments are inserted. Aside from proving a point, this data would be useful to have in hand to serve as a performance benchmark for second impairments.

5.4. *Probability Smoothing and Quantization*

As explained in Chapter 4, smoothing and quantizing the probabilities was thought to be a necessary step towards extracting an accurate and stable estimate for MMAE bank swapping and Control Reconfiguration purposes. For this thesis research, the quantizer levels were set to 0.95, 0.9, 0.75, 0.5, and 0.0 probability levels, and *all levels* on *all* 12 MMAE channels had leading edge and trailing edge timers set to 0.15 and 0.5 seconds, respectively. It is suggested that, due to the initial tendency of the probabilities to be much higher than the actual partial actuator impairment, quantization levels may have to be biased up or down with respect to actual probability values to aid in proper filter bank swapping within MMAE, as well as to provide an accurate estimate to the control reconfiguration module. It may also be necessary to adjust the leading and/or trailing edge timers to speed up or desensitize the bank swapping logic to probability spikes or drop-outs. It is strongly suggested, however, that one explore *effectively biasing* (Section 5.2.3) the filter bank swapping logic *before* modifying the quantizer levels or time constants.

5.5. *MMAE Filter Bank Swapping*

As explained in Chapter 4, prior research efforts were able to trigger filter bank swapping by using a simple probability threshold "latch". In this thesis effort, filter bank swapping for partial actuator impairments is totally dependent on the accuracy of the first estimate, which is an output of the probability quantizer. It has been determined that incorrect bank swapping has probably occurred for the majority of cases and has adversely affected results. Since the possibility exists that the quantization scheme may never perform to expectations, a backup plan is offered. The entire probability smoothing and quantization

scheme may become unnecessary if it is decided that partial impairment Kalman filters for every actuator will be included in "Level 0", or within intermediate filter banks synthesized for partial first impairment resolution. Another possibility is to utilize partial failure Kalman filters for accurate estimation of the first impairment and for correct bank swapping, and to use the [less accurate and less complicated] smoothing and quantization scheme to estimate the second partial impairment.

5.6. *Control Reconfiguration Performance*

This thesis was concerned primarily with studying MMAE's ability to estimate single and dual, partial actuator impairments during non-maneuver conditions, with control reconfiguration "in the loop". It was due primarily to the absence of large amplitude maneuvers that an evaluation of the ability of Control Reconfiguration to restore lost control authority was not practical. This should not be of great concern, since prior research by Stepaniak and Lewis [26, 45], as well as much earlier simulation research [7, 11, 12, 15, 22, 38] has long since established that Control Reconfiguration does indeed restore lost control authority *if it is supplied with an accurate identification and estimate of the partial impairment.* Even though Control Reconfiguration performance was not the major topic of this research effort, a characteristic of Control Reconfiguration, as it is implemented in this research, was noticed and needs to be addressed.

The results of this thesis question the validity of reconfigured actuator commands during total impairments of either the left or right flaperon. When examining state plots for cases of a total flaperon impairment (Appendix E.2), it is seen that the other flaperon is "frozen" at a constant deflection, as if it too were totally impaired. This happens even when a roll doublet is commanded. The other, healthy flaperon is "zeroed", even though it is the primary means of roll control, and the stabilators are issued the necessary increased differential commands. Lewis [26] noticed this and correctly surmised that, since the pre-packed reconfiguration matrix ($\mathbf{D}_{a3}$ or $\mathbf{D}_{a4}$) corresponding to either flaperon impairment will contain a "1" in the row and column corresponding to the healthy flaperon, then the flaperon commands will cancel mathematically. Control Reconfiguration will send the command meant originally for the impaired flaperon to the unimpaired flaperon, which makes complete sense, but when that command is summed with the

existing command (of opposite sign) being sent to the healthy flaperon, the net effect is zero, which does not make practical sense at all! From intuition as well as from prior research experience [11, 12] one expects the healthy flaperon to receive roughly twice its normal deflection, which will happen, of course, if the sign on the "1" is reversed. For example, given the $\mathbf{D}_{a3}$ (the left flaperon fully failed) reconfiguration matrix, the suggested correction is illustrated below:

Given:

$$\mathbf{D}_{a3} = \begin{bmatrix} 1 & 0 & 1.1037 & 0 & 0 \\ 0 & 1 & -1.1037 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0.8678 & 0 & 1 \end{bmatrix} \tag{5-1}$$

The suggested correction is:

$$\mathbf{D}_{a3}(4,3) = -\mathbf{D}_{a3}(4,3) \tag{5-2}$$

$$= -1$$

One should not, however, proceed to reverse these signs without demonstrating that an error exists in the original development. Two possible and closely related sources of error are 1) a misunderstanding of the sign conventions used for flaperons, or 2) an incorrect synthesis of the linear F-16 model, and the **B** matrix in particular, since it is the **B** matrix which impacts the control reconfiguration calculations directly. If no error can be found, then it may be necessary simply to argue the point that a commanded deflection to one flaperon yields equivalent results (in a linear sense) to a commanded deflection in the opposite direction to the opposite flaperon, and therefore the signs in $\mathbf{D}_{a3}$ and $\mathbf{D}_{a4}$ should be changed. It is strongly recommended that this disagreement between mathematical, control reconfiguration calculations and common sense (as well as prior research experience) be resolved. Until it is resolved, the full benefits accruing from Control Reconfiguration in this application will not be realized, and one may be misled into believing that it would be better to "do without" control reconfiguration.

5.7. *Recommended Check-Cases and Modifications to the Simulation*

5.7.1. *Recommended Check-Cases*

It is *highly* recommended that two important check-cases be made, for confidence purposes, in the overall simulation. The first is a recommended comparison between the linear, time-invariant F-16 model used by the Kalman filter hypothesizing a "fully functional" aircraft, and the VISTA F-16 truth model at the design condition of Mach .4 and 20000'. This comparison would be easy to accomplish and involves simply inserting the code necessary to "pick off" and record the linear model aircraft states, $\mathbf{x}_{aug}[1,..8]$, to an output file. The simulation would then be run without impairments inserted, with modeling and measurement noises turned off, and with dithering on, and the state outputs from the linear model would be compared to (subtracted from) the same states as output from the truth model. This comparison will yield important insight on sources of error inherent in the linear model, and may shed appropriate light on the strong periodic behavior seen in the Probability Summary Plots for cases of partial impairment.

The second recommended check-case could be the most important suggestion made in this chapter, and that is to compare state outputs from a Kalman filter, in the MMAE bank and hypothesizing a partially failed actuator, to the same state outputs as output from the F-16 truth model with the same partial impairment. It was stated in Chapter 4 that the method of inserting partial failures into the F-16 truth model was selected to be "what the MMAE algorithm was tuned to see", and that it gave identical results for cases of "fully functional" and for "fully impaired" actuators. Earlier thesis efforts implemented total actuator impairments by forcing the impaired actuator's position to its trim deflection, hence, agreement was guaranteed between the impaired, linear perturbation model within the Kalman filter and the truth model, and good research results were obtained. It may be that the truth model partial actuator impairment model and the actuator partial impairment as modeled by the Kalman filter are not in close enough agreement (due to aircraft trim bias) at points in-between fully functional and fully failed, and this suggested check-case should make any discrepancies apparent.

5.7.2. *Recommended Modifications to the Simulation*

Inserting code to calculate the actual pseudoinverse used by Control Reconfiguration, is strongly recommended. Stepaniak [45] utilized a pre-packed reconfiguration matrix for his study of single, total impairments which worked very well and greatly decreased computational loading. Lewis [26] tried the same approach in his study of dual, total impairments, but had to pre-compute and store 15 pre-packed matrices corresponding to the number of possible pairs of actuator total impairments. This research used a two-stage method using computations equivalent to the true inverse for dual, partial impairments. Since future research will no doubt study combinations of partial and total actuator impairments, it is recommended that *one* approach to computing reconfigured actuator commands be adopted, and that is to use the actual pseudoinverse.

A suggestion is also made to add a 13th subplot to the Probability Summary Plots to display the value of the MMAE filter bank as a function of time. Current methods of tracking filter bank switching involve outputting messages displayed on the terminal to a file and then examining the contents. This creates huge files over the course of hundreds of Monte Carlo runs which quickly overwhelm storage capacity. These output files would be processed exactly like probability information (with mean values and standard deviations) to get plots, and would allow more accurate analysis of Probability Summary and regular Probability plots. Another option is to simply compare the "desired Level '1'filter bank given the a-priori known first impairment" and set an error flag (which would be written to an output file) if a wrong bank switch is made.

5.8. *Recommended Directions for Future Research: Near and Far Term*

5.8.1. *Near Term*

This research effort, while originally thought to be merely a continuation of previous research efforts, became more of a ground-breaking feasibility study. As such, many more issues and problems were raised, as well as answering some questions. It is therefore recommended that this research be duplicated and continued in the future, but after some of the questions raised by this thesis have been answered. Parallel research efforts should include expansion of the MMAE flight envelope for total, single actuator failures to be followed by dual, total failures. The desired outcome would be demonstration of the ability by MMAE to detect those total failures with the same performance as it currently does at a discrete point in the flight envelope. Envelope expansion might include "scheduling" the linear perturbation models inside the Kalman filters with changing flight condition, or recording and then scheduling the truth model trim deflections with flight condition, or both. This would enable investigation of algorithm performance during vehicle maneuvering (rather than just with control dithering with no maneuver commands), which could not be attempted with filters designed only for a single point in the flight envelope.

5.8.2. *Far Term: Man-in-the-Loop Simulation*

Real-time, man-in-the-loop simulation should be considered *after* the operable MMAE envelope has been successfully expanded to large enough bounds so that a human operator, or at least an autopilot, can remain within the valid operating range. The only real-time, man-in-the-loop task which is practical at this time is an evaluation of Control Reconfiguration "stubbed" with perfect failure status (and state) information, i.e., with the MMAE turned off. The pilot might be asked to fly the impaired aircraft behind a maneuvering target while switching between Control Reconfiguration and normal Block 40 commands, and to compare the results, which should be fairly dramatic.

5.9. *Chapter Summary*

Clearly, the MMAE algorithm as it is currently implemented has *some* ability to blend hypotheses and to estimate a partial actuator impairment. Performance is not as good as expected or hoped, however. The task is now to improve MMAE performance during detection of *single, partial* actuator impairments: in particular, to get an accurate estimate (from the quantizer or from some other scheme) in 1 second or less, and ensure that MMAE transitions to the correct "Level 1" filter bank. Suggestions to achieve this performance are presented in this chapter, along with special methods to detect rudder impairments. Once these initial performance problems are tackled and solved, then MMAE detection and estimation performance during *dual*, partial actuator impairments will improve. Suggestions were made for improving Control Reconfiguration performance, as well as improving the thesis research environment. Suggestions were also made for areas of parallel research, which would not depend directly upon the results of this thesis. Suggested topics were: MMAE envelope expansion, and starting with single total actuator impairments and progressing to dual, total actuator impairments. Man-in-the-loop simulation was presented as a far-term research objective.

Appendix M An Alternative Control Reconfiguration Method

For Cases of Partial Actuator Impairment

In cases of partial actuator impairment ($0<\varepsilon<1$), the direct pseudoinverse calculation (Equation (3-35)) is equal to the true inverse calculation: $\mathbf{F}_{ai}^{-1}$. The true inverse calculation will command the partially impaired actuator to $1/\varepsilon$ times the original command, and, as ε assumes values less than 0.5 (50% actuator failure or greater) this may cause actuators to be commanded to rate and/or position saturations. Maybeck [37] has suggested an alternative to computing and using $\mathbf{F}_{ai}^{-1}$ (Section 3.9.2), in cases of partial impairments, which would simplify redistribution calculations and *perhaps* prevent rate and position saturations of remaining actuators.

The alternative to using the true inverse calculation, $\mathbf{F}_{ai}^{-1}$, for *partially* impaired actuators is to use a variant of the pre-packed $\mathbf{D}_{ai}$ matrix (Equation (3-37)) for *totally* failed actuators. Given a partial impairment of the i^{th} actuator such that its effectiveness is ε, the procedure is to multiply all j elements ($j \neq i$) in column 'i' of $\mathbf{D}_{ai}$ by ε. The command to the impaired actuator ($j = i$) is equal to 1.0, and all *other* columns of "$\mathbf{D}_{ai}$ variant" contain ones on the diagonal. It can be proven [49] that this alternative method satisfies Equation (3-32). An example is given for the case of a 70% left stabilator (i=1) actuator impairment (actuator effectiveness, ε =.3). The desired condition is expressed by Equations (3-30) and (3-32):

$$\mathbf{B}_{fail}\mathbf{u}_r \approx \mathbf{B}\mathbf{u} \tag{3-30}$$

$$\mathbf{u}_r = \mathbf{D}_{ai}\mathbf{u} \tag{3-32}$$

For the impairment case specified, Equation (3-36) applies and results in:

$$\mathbf{D}_{ai} = \mathbf{F}_{ai}^{-1} = \begin{bmatrix} 3.3 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The alternative method for this case is to take the fully-failed Control Reconfiguration matrix, $\mathbf{D}_{a1}$, for the left stabilator:

$$\mathbf{D}_{a1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ +0.906 & 0 & 1 & 0 & 0 \\ -0.906 & 0 & 0 & 1 & 0 \\ +0.786 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Entry (1,1) is set equal to 1.0 and all other entries in the first column (column $i = 1$) are multiplied by 0.3 ($\varepsilon = .3$), to yield:

$$\mathbf{D}_{a1\,variant} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0.3 & 1 & 0 & 0 & 0 \\ +0.2718 & 0 & 1 & 0 & 0 \\ -0.2718 & 0 & 0 & 1 & 0 \\ +0.2358 & 0 & 0 & 0 & 1 \end{bmatrix}$$

This result inserted into Equation (3-32) satisfies Equation (3-30), and may prevent the left stabilator from being "overdriven" into rate or position saturation.

1. Athans, M., *et al,* "The Stochastic Control of the F-8C Aircraft Using a Multiple Model Adaptive Control (MMAC) Method - Part I: Equilibrium Flight," *IEEE Transactions on Automatic Control, AC-22(5)*:768-780 (October 1977).

2. Baram, Y., "Information, Consistent Estimation and Dynamic System Identification," Air Force Office of Scientific Research Technical Report, Number 77-0283, available from the Defense Technical Information Center, AD-A037972, November 1976.

3. Baram, Y., and Sandell, N.R., Jr., "An Information Theoretic Approach to Dynamic System Modeling and Identification," *IEEE Trans. Automat. Control* **AC-23** (1), 61-66 (1978).

4. Baram, Y., and Sandell, N.R., Jr., "Consistent Estimation of Finite Parameter Sets with Application to Linear Systems Identification," *IEEE Trans. Automat. Control* **AC-23** (3), 451-454 (1978).

5. Blakelock, John H., *Automatic Control of Aircraft and Missiles*. New York: John Wiley & Sons, Inc., 1991.

6. Caglayan, A. K. *et al*, "Detection, Identification, and Estimation of Surface Damage / Actuator Failure for High Performance Aircraft," *Proceedings of 1988 ACC*, Atlanta, GA., June 1988.

7. Chandler, P.R., "Reconfigurable Flight Control at Wright Laboratory," technical note prepared for AGARD Flight Vehicle Integration Panel 2020 report, available from Wright Laboratory / WL / FIGC, Wright-Patterson AFB, OH.

8. Chandler, P.R., and Rubertus, D.P., "A Systems Approach to Flight Control Reliability and Maintainability," unpublished technical note available from the Wright Research and Development Center, Flight Dynamics Laboratory, Control Systems Development Branch, Wright-Patterson AFB, OH.

9. Chang, C.B., and Athans, M., "State Estimation for Discrete Systems with Switching Parameters," *IEEE Transactions on Aerospace and Electronic Systems, AES-14(4)*:418-424 (May 1978).

10. Chang, C.B., and Whiting, R.H., and Athans, M., "On the State and Parameter Estimation for Maneuvering Re-entry Vehicles," *IEEE Trans. Automat. Control* **AC-22** (1), 99-105 (1977).

11. Clark, C.S., Anderson, J.A., Madsen, P., and Unfried, F., "The AFTI F-16 Control Mixer Reconfiguration Simulation," Unpublished, *NAECON (1987)*

12. Clark, C.S., U.S. Air Force Wright Aeronautical Laboratories, Wright-Patterson AFB, OH. Personal Knowledge and Participation. 1985-1992.

13. Cox, H., "On the Estimation of the State Variables and Parameters for Noisy Dynamic Systems," *IEEE Trans. Automat. Control*, **AC-9** (1), 5-12, (1964).

14. Eide, Capt Peter K., *Implementation and Demonstration of a Multiple Model Adaptive Estimation Failure Detection System for the F-16*. MS Thesis, AFIT/GE/ENG/94D-06, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH., December 1994.

15. Eslinger, R. A. and Chandler, P.R., "Self-Repairing Flight Control System Program Overview," Air Force Wright Aeronautical Laboratories Technical Report, Number 88-0379, Wright-Patterson AFB, OH 1988.

16. Fry, C.M. and Sage, A.P., "On Hierarchical Structure Adaptation and Systems Identification," *International Journal of Control, 20(3):*433-452 (September 1974).

17. General Dynamics, Fort Worth Division, "Dwg 711ZC001 - Rev A," September 1990.

18. General Electric Company, *Self Repairing Digital Flight Control System Study*, AFWAL-TR-88-3007.

19. Greene, C.S., *An Analysis of the Multiple Model Adaptive Control Algorithm*, Ph.D. dissertation, ESL-TH-843, Elec. Systems Lab., MIT, Cambridge, Massachussetts (August 1978)

20. Greene, C.S., and Willsky, A.S., "An Analysis of the Multiple Model Adaptive Control Algorithm," *Proc. IEEE Conf. Decision and Control, Albuquerque, New Mexico* pp. 1142-1145 (December 1980)

21. Grumman Aerospace Corporation, *Dispersed and Reconfigurable Digital Flight Control System*, AFFDL-TR-79-3125, December 1979.

22. Grumman Aircraft Corporation, *Control Reconfigurable Combat Aircraft Development*, Vol I-IV, AFWAL-TR-88-3118, December 1989.

23. Inc., Century Computing, *Simulation/Rapid-Prototyping Facility (SRF) F-16 VISTA Simulation Engineer's Guide*, December 1992.

24. Lainotis, D.G., "Partitioning: A Unifying Framework for Adaptive Systems, I: Estimation," *Proceedings of the IEEE*, 64:1182-1197 (August 1976).

25. Lane, Capt David W., *Multiple Model Adaptive Estimation Applied to the LAMBDA URV for Failure Detection and Identification*. MS Thesis, AFIT/GE/ENG/93D-23, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH., December 1993.

26. Lewis, Capt Robt. W., *Multiple Model Adaptive Estimation and Control Redistribution for the Vista F-16*, MS Thesis, AFIT/GE/ENG/96D-29, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH., December 1996.

27. Magill, D.T., "Optimal Adaptive Estimation of Sample Stochastic Processes," *IEEE Conference on Decision and Control*, AC-10(4):434-439 (October 1965).

28. Martin, Capt Richard M., *LQG Synthesis of Elemental Controllers for AFTI/F-16 Adaptive Flight Control*. MS Thesis, AFIT/GE/ENG/90D-36, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH., December 1990.

29. Maybeck, P.S. and Hentz, K P., "Investigation of Moving-Bank Multiple Model Adaptive Algorithms," *Journal of Guidance and Control*, 10(1):90-96 (January-February 1987).

30. Maybeck, P.S., and Suizu, R.I., "Adaptive Tracker Field of View Variation Via Multiple Model Filtering," *IEEE Transactions on Aerospace and Electronic Systems, 21(4)*:529-537 (July 1985).

31. Maybeck, P.S. and Hanlon, Capt P.D., "Performance Enhancement of a Multiple Model Adaptive Estimator," *IEEE Conference on Decision and Control* (1993).

32. Maybeck, P.S. and Pogoda, D.L., "Multiple Model Adaptive Controller for the STOL F-15 with Sensor / Actuator Failures," *Proceedings of the 28th Conference on Decision and Control*, 1566-1572 (December 1989).

33. Maybeck, P.S. and Stevens, R.D., "Reconfigurable Flight Control Via Multiple Model Adaptive Control Methods," *IEEE Transactions on Aerospace and Electronic Systems*, 470-480 (May 1991).

34. Maybeck, P.S., *Stochastic Models, Estimation, and Control, I.* New York: Academic Press, Inc., 1979.

35. Maybeck, P.S., *Stochastic Models, Estimation, and Control, II.* New York: Academic Press, Inc., 1982.

36. Maybeck, P.S., *Stochastic Models, Estimation, and Control, III.* New York: Academic Press, Inc., 1982.

37. Maybeck, P.S., Professor, Department of Electrical and Computer Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH. Personal Interviews. April-December, 1997.

38. McDonnell Aircraft Company, *Self Repairing Flight Control System, Vol I: Flight Test Evaluation on an F-15 Aircraft*, WL-TR-91-3025, August 1991.

39. Menke, T.E. and Maybeck, P.S., "Sensor / Actuator Failure Detection in the VISTA F-16 by Multiple Model Adaptive Estimation," *Proceedings of American Control Conference*, San Francisco, June 1993 (1993).

40. Menke, T.E., *Multiple Model Adaptive Estimation Applied to the VISTA F-16 with Actuator and Sensor Failures.* MS Thesis, AFIT/GA/ENG/92J-01, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH., June 1992.

41. MIL-STD-1797A, *Flying Qualities of Piloted Aircraft*, January 1990.

42. Moore, J. B. and Hawkes, R. M., "Decision Methods in Dynamic System Identification," *Proceedings of the IEEE Conference on Decision and Control 14.* 645-650. 1975.

43. Nelson, R.C., *Flight Stability and Automatic Control.* New York: McGraw-Hill, Inc., 1989.

44. Pogoda, Capt D.L., *Multiple Model Adaptive Controller for the STOL F-15 with Sensor / Actuator Failures.* MS Thesis, AFIT/GE/ENG/88D-23, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH., December 1988.

45. Stepaniak, 2nd Lt. M.J., *Multiple Model Adaptive Control of the VISTA F-16.* MS Thesis, AFIT/GE/ENG/95D-04, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH., December 1995.

46. Stevens, Capt R.D., *Characterization of a Reconfigurable Multiple Model Adaptive Controller Using A STOL F-15 Model.* MS Thesis, AFIT/GE/ENG/89D-52, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH., December 1989.

47. Strang, G., *Linear Algebra and Its Applications* (third Edition). San Diego: Harcourt Brace Jovanovich, Publishers, 1988.

48. Stratton, Capt G.L., *Actuator and Sensor Failure Identification Using a Multiple Model Adaptive Technique for the VISTA / F-16* . MS Thesis, AFIT/GE/ENG/91D-53, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH., December 1991.

49. The MathWorks, Inc., *MATLAB* (registered trademark) (Version 4.2c Edition), November 1994.

50. Willsky, A. S., "A Survey of Design Methods for Failure Detection in Dynamic Systems," *Automatica*, 12:601-611, 1976.

51. WL/FIGXF, WPAFB, OH., "VISTA / NF-16 Technical Data for Customer Usage," August 1991.

Vita

Curtis Steven Clark was born in Hyannis, Massachussetts, on August 20, 1959. He grew up in Fairborn, Ohio, acquired his pilot license while in high school, and entered the U.S. Army in 1977, where he served as an Air Traffic Controller. He entered Wright State University in 1980, graduated in 1985 with a B.S. Degree in Engineering-Physics (cum laude), and was inducted into Sigma Pi Sigma Physics Honor Society the same year. He started work at Air Force Wright Aeronautical Laboratories in 1985, and married his wife, Janice Mensi, in 1986. He worked as a Flight Simulation Engineer until winning a DAGSI scholarship in 1996 to attend the AFIT School of Electrical Engineering full-time to pursue his Master of Science Degree in Electrical Engineering. Following graduation, Curtis will return to Wright Laboratories, but will also return to his second vocation of Flight Instructor (Airplane). He plans to pursue his Ph.D. in the near future, and would enjoy teaching others in his profession.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1997	Master's Thesis

4. TITLE AND SUBTITLE
MULTIPLE MODEL ADAPTIVE ESTIMATION AND CONTROL REDISTRIBUTION PERFORMANCE ON THE VISTA F-16 DURING PARTIAL ACTUATOR IMPAIRMENTS

5. FUNDING NUMBERS

6. AUTHOR(S)
Curtis S. Clark
GS - 13, D.O.D.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-6583
Capt. Odell Reynolds
WL/FIGS
2210 Eighth St. STE 11
Wright-Patterson AFB, OH 45433-7521

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GE/ENG/97D-23

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for public release; Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Multiple Model Adaptive Estimation with Control Reconfiguration (MMAE/CR) capability to estimate and compensate for partial actuator failures, or "impairments" is investigated using the high-fidelity, nonlinear, six-degree-of-freedom, VISTA F-16 simulation which currently resides on the Simulation Rapid-Prototyping Facility (SRF). After developing a model for inserting partial actuator impairments into the VISTA F-16 truth model, research begins with a battery of single actuator impairment tests. This stage of research explores the capability of the existing MMAE algorithm to estimate single, partial actuator impairments, and helps to define refinements and expansions needed in the MMAE algorithm for the second phase of research: the detection and estimation of dual, total and partial actuator impairments. It is seen from the first stage of research that, while MMAE is able to estimate partial impairments, there are refinements needed, such as "probability smoothing and quantization", to compensate for the quality of MMAE probability data and to provide a better, more stable estimate value to the Control Reconfiguration module. The Kalman filters and the dual, partial failure filter banks necessary for the detection of dual, partial actuator impairments are also defined as a result of the single impairment tests. Fifteen more banks of "partial first-failure" Kalman filters are added to the existing MMAE algorithm, as well as the "bank swapping" logic necessary to transition to them. Once the revised and expanded MMAE/CR algorithm is ready, research begins on dual combinations of total and partial actuator impairments. While results of these tests (for other than total impairments) are not as good as originally hoped or expected, the potential for better performance is evident.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Multiple Model Adaptive Estimation, MMAE, Kalman Filter, F-16, Control Reconfiguration, Flight Control, Failure Detection, Reconfigurable Control	400
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL